

AÑO II N.º 52

**MICROHOBBY**

# AMSTRAD

*Semanal*

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

AÑO II N.º 52

**160 Ptas.**

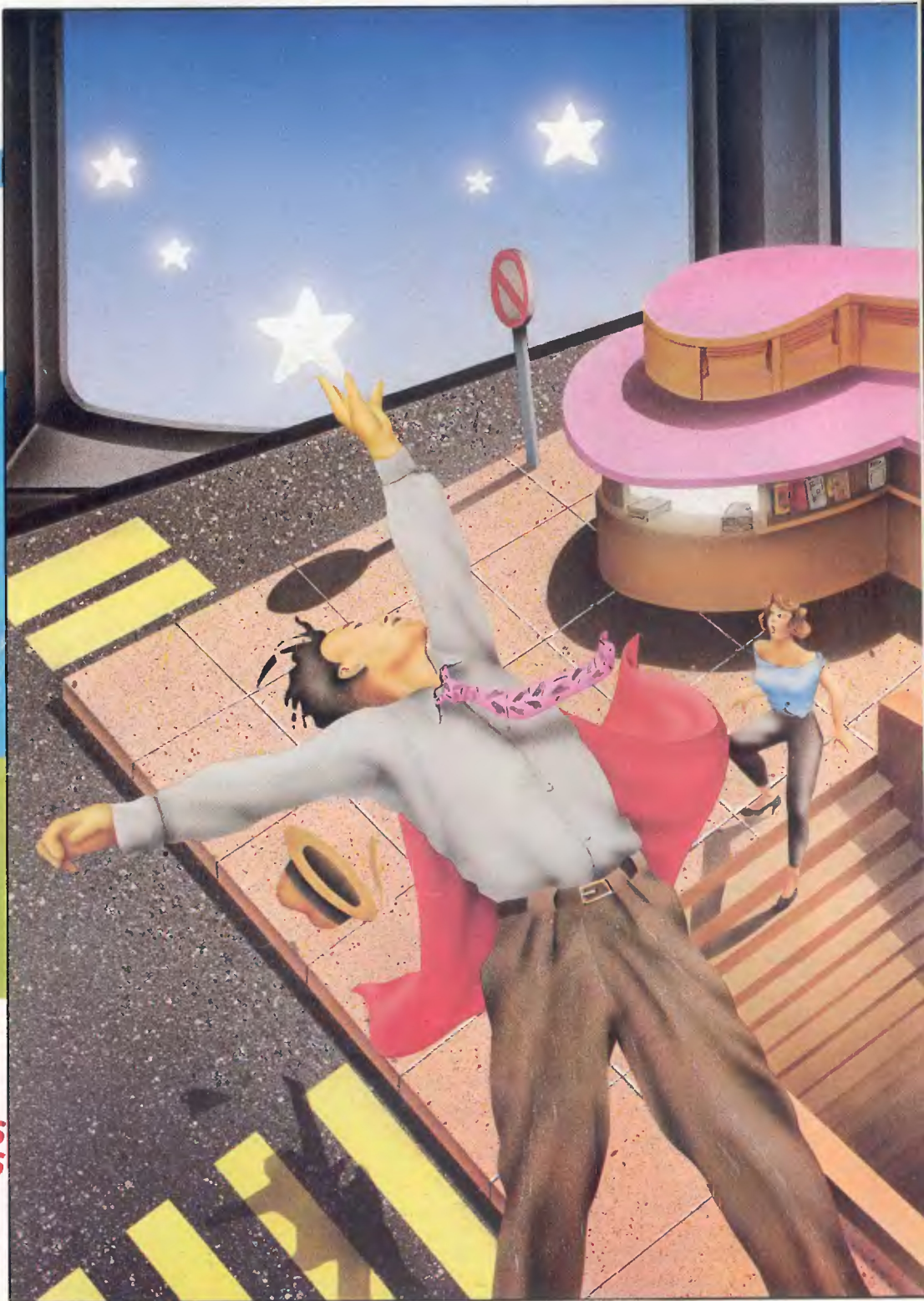
Canarias 165 pts.

**PINCELES HECHOS  
DE BITS (Y II)**

**CONOCE EN  
TODO  
MOMENTO  
LO QUE SE  
OCULTA EN EL  
CORAZON DE  
TU AMSTRAD**

**EL SABOR DE LA  
AVENTURA DE LA  
MANO DE  
GREEN BERET**

**UN NUMERO  
ILIMITADO DE TIPOS  
DE LETRA PARA TUS  
PROGRAMAS**



**HOBBY PRESS**



Versión  
SPECTRUM,  
AMSTRAD Y COMMODORE

# Alistate a **Juegos & ESTRATEGIA**

## LA BATALLA DE INGLATERRA ha comenzado

Pilota tu «Spitfire»  
sobre el Canal de La Mancha  
y defiende el  
destino  
del mundo libre.



Oferta especial  
hasta el 31  
de noviembre:  
PIDE TRES NUMEROS  
Y PAGA  
SOLO DOS.



ENVIE HOY MISMO ESTE CUPON AL APARTADO 232 DE ALCOBENDAS (Madrid)

- ☐ Deseo recibir en mi domicilio tres ejemplares de **Juegos & Estrategia** al precio especial de 2.255 ptas., lo que me supone adquirir tres y pagar sólo dos. Marco los tres ejemplares que deseo con una cruz.
- ☐ Deseo recibir un solo ejemplar de **Juegos & Estrategia** al precio de 1.125 ptas. Marco con una cruz el ejemplar que deseo recibir.

### Spectrum

- N.º 1 ☐ Arnhem  
N.º 2 ☐ Ratos del Desierto  
N.º 3 ☐ OTAN Alerta  
War Zone  
Especial 1 ☐ Elecciones Generales  
N.º 4 ☐ Su mejor hora (La batalla de Inglaterra)

### Amstrad

- ☐ Arnhem  
☐ Ratos del Desierto  
☐ Teatro de Europa  
War Zone

### Commodore

- ☐ Teatro de Europa

- ☐ La batalla de Inglaterra ☐ La batalla de Inglaterra

NOMBRE

DIRECCION

LOCALIDAD

C. POSTAL

TELEFONO

PROVINCIA

PROFESION

Fecha de  
nacimiento

Forma de pago:

☐ Talón bancario a nombre de Hobby Press, S. A. ☐ Giro Postal a nombre de Hobby Press, S. A., n.º de giro

☐ Tarjeta de crédito: Visa n.º

Master Charge n.º

American Express n.º

Fecha de caducidad de la tarjeta

Fecha y firma



# AMSTRAD

## sumario

**Director Editorial**  
José I. Gómez-Centurión

**Director Ejecutivo**

José M.º Díaz

**Redactor Jefe**

Juan José Martínez

**Diseño gráfico**

Fernando Chaumel

**Colaboradores**

Eduardo Ruiz

Javier Barceló

David Sopuerta

Robert Chatwin

Francisco Portolo

Pedro Sudón

Miguel Sepúlveda

Francisco Martín

Jesús Alonso

Pedro S. Pérez

Amelio Gómez

Alberto Suñer

**Secretaría Redacción**

Carmen Sontamaria

**Fotografía**

Carlos Condel

Chema Sacristán

**Portada**

Javier Igual

**Ilustradores**

J. Igual, J. Pons, F. L. Frontán,

J. Septien, Pejo, J. J. Mora

**Edita**

HOBBY PRESS, S.A.

**Presidente**

María Andrión

**Consejero Delegado**

José I. Gómez-Centurión

**Jefe de Producción**

Carlos Peropadre

**Marketing**

Marta García

**Jefe de Publicidad**

Concha Gutiérrez

**Publicidad Barcelona**

José Galán Cortés

Tel: (93) 303 10 22/313 71 62

**Secretaría de Dirección**

Marisa Cogorro

**Suscripciones**

M.º Rosa González

M.º del Mor Colzado

**Redacción, Administración y Publicidad**

Ctra. de Irún km 12,400

(Fuencarral) 28049 Madrid

Teléfonos: Suscrip.: 734 65 00

Redacción: 734 70 12

**Dto. Circulación**

Paulino Blanco

**Distribución**

Coedis, S. A. Valencia, 245

Barcelona

**Imprime**

ROTEC, S. A. Ctra. de Irún,

Km. 12,450 (MADRID)

**Fotocomposición**

Novocomp, S.A.

Nicolás Morales, 38-40

**Fotomecánica**

GROF

Ezequiel Solana, 16

**Depósito Legal:**

M-28468-1985

**Derechos exclusivos**

de la revista

**COMPUTING with**

**the AMSTRAD**

Representante para Argentina, Chile,

Uruguay y Paraguay, Cia.

Americana de Ediciones, S.R.L. Sud

AMERICA 1.532. Tel.: 21 24 64, 1209

BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace

necesariamente solidaria de las

opiniones vertidas por sus

colaboradores en los artículos

firmados. Reservados todos los

derechos.

Año II • Número 52 • 9 al 15 de Septiembre  
160 ptas. (incluido I.V.A.)  
Canarias, 155 ptas. + 10 ptas. sobretasa aérea.  
Ceuta y Melilla, 155 ptas.

## ProgramAcción 10

Muchos ordenadores, como el Apple Macintosh y el Commodore Amiga, tienen incorporada la posibilidad de usar distintos tipos de letra, itálica, negrita, etc.

**Amstrad** no va a ser menos, y os presentamos un programa que va a permitir obtener un número prácticamente ilimitado de tipos de letra distintos, utilizables en todos los programas.



## Mr. Joystick 14

**Green Beret** es un gran juego. Divertido, adictivo y espectacular, por su concepción y por sus gráficos. Vais a disfrutar mucho, pero que mucho, por él.



## Util al programador 16

**Micro** es un programa, hecho en Basic, que te va a permitir saber en cualquier momento el contenido de todos los registros del microprocesador de tu **Amstrad**.



## Código máquina 22

Continuando con el artículo de la semana pasada, sólo nos queda averiguar cómo se manejan en lenguaje máquina las órdenes gráficas de movimiento relativo, es decir, el equivalente Basic de PLOT, DRAW, etc.

## Serie Oro 28

A los que les apasionen las máquinas «de bolitas», que los ingleses llaman «Pinball», a partir de ahora lo tienen muy claro. Podrán jugar hasta hartarse con su **Amstrad** precisamente con un programa llamado Pinball.



# AMSTRAD DMP 2000

## NO ENCONTRARA UNA IMPRESORA QUE LE HAGA TAN BUEN PAPEL.



Soportes abatibles que permiten colocar el papel bajo la impresora.

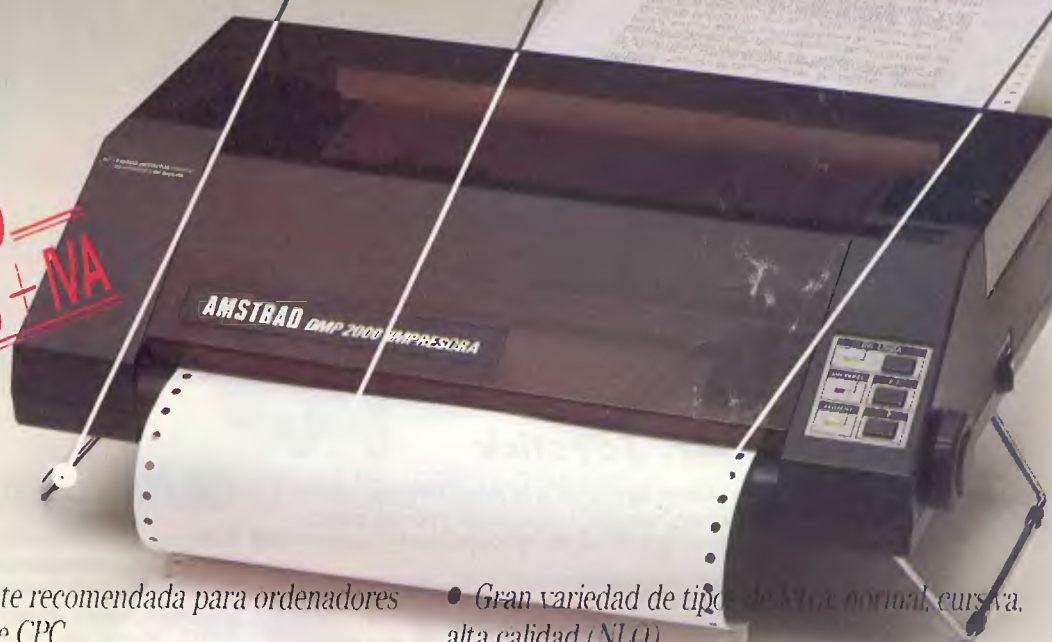


Cómodo sistema de carga frontal del papel.



Admite diferentes anchos de papel, tanto continuo (de 114 a 254 mm.) como hojas sueltas (102 a 241 mm.)

**POR SOLO  
39.500 PTAS + IVA**



- Especialmente recomendada para ordenadores AMSTRAD serie CPC.
- Conectable a cualquier ordenador con interface centronics.
- Velocidad de impresión de 105 caracteres por segundo.

- Gran variedad de tipos de letra normal, cursiva, alta calidad (NLQ)
- 40, 66, 80 y 132 caracteres por columna.
- Impresión de gráficos punto a punto en diferentes densidades.
- 96 caracteres ASCII y 8 subconjuntos internacionales.

*¡¡ Increíble !!*

**AMSTRAD** ESPAÑA

GRUPO INDESCOMP



# SOFTWARE EDUCATIVO

## LOS LIBROS QUE VIENEN

Primera  
PLANA

**D**imensión **NEW** nos ha sorprendido con un programa, que ellos llaman integrado, al cual podríamos encuadrarlo, entre otros campos posibles, dentro del software educativo. El paquete versa sobre matemáticas, e incluye una serie de subprogramas de utilidad para resolver los problemas de más de un estudiante, si no con mucha espectacularidad, sí al menos con concisión y eficacia, hasta tal punto, que más de un usuario aficionado a florituras lo calificaría de seco.

Nosotros nos reservamos nuestra opinión hasta que se le someta a un banco de pruebas serio, el cual revelará las verdaderas posibilidades del paquete, imposibles de apreciar a simple vista.

Pero los ávidos de detalles, ansiedad muy lógica por otra parte, el espectro matemático del paquete es el siguiente:

- Funciones algebraicas.
- Funciones trascendentes.
- Cálculo matricial y sistemas de ecuaciones lineales.
- Ecuaciones diferenciales.
- Resolución mediante series de Fourier.

El programa está distribuido por IDEALOGIC, que está en:  
C/ Valencia 85, Barcelona.  
Tels.: 253 86 93 y 253 89 09.



**E**ditorial ANAYA va publicar próximamente dentro de la colección TEMAS UNIVERSITARIOS CON MICROORDENADOR los siguientes libros:

— **ESTADISTICA.** Este libro nos narra los importantes avances que han realizado los especialistas con la utilización de ordenadores.

Y aunque estos ordenadores funcionan normalmente con lenguajes como el **FORTRAN, PASCAL** y otros matemáticos más especializados, quedan todavía muchas posibilidades al Basic con los programas de probabilidad y estadística.

Los programas que tiene el libro emplean el Basic extendido disponible habitualmente en las grandes unidades centrales.

Autor: J. Tennant-Smith.

Precio: 1.643 ptas.

Comercializa: Grupo Distribuidor Editorial.

— **TERMODINAMICA Y TRANSMISION DE CALOR.** La facilidad y fiabilidad, así como la rapidez en el cálculo que consiguen los ordenadores en el campo de la ingeniería, hacían necesario un libro que ayudará al ingeniero a desarrollar su habilidad en el manejo de esta importante herramienta.

Este libro pretende ayudar al lector a utilizar la programación en Basic y aprovechar la ventaja de ser capaz de estudiar el efecto que se produce sobre los resultados de un programa, al cambiar los valores de los datos de entrada, cosa ésta interesantísima y que debe convencer al ingeniero del valor de los ordenadores en los cálculos de diseño y optimización.

Autor: D. H. Bacon.

Precio: 1.749 ptas.

Comercializa: Grupo Distribuidor Editorial.

— **METODOS MATRICIALES.** Este libro constituye un medio por el que nos podemos introducir en los métodos matriciales y sus aplicaciones a la resolución de sistemas de ecuaciones lineales y a la aproximación y ajuste de datos numéricos.

Después de describir algunos fundamentos matemáticos se dan los algoritmos y programas en Basic para resolver los métodos matriciales propuestos.

El libro contiene gran cantidad de información útil, de forma que se pueda utilizar por estudiantes de ciencias e ingeniería, así como por científicos y estadísticos.

También permite el aprendizaje del Basic mediante ejemplos prácticos y muestra cómo se puede aplicar la computación para la resolución de problemas reales.

Autor: J. C. Mason

Precio: 1.643 ptas.

Comercializa: Grupo Distribuidor Editorial.

Además, esta **SYMPHONY** es un manual y está pensado como fuente de información rápida y fácil de usar para el paquete de modelado LOTUS SYMPHONY.

No aspira a ser exhaustivo ni sustituir a los manuales **SYMPHONY**, pero resulta de gran utilidad para el lector, por la claridad y sentido pedagógico con que está escrito, ya que fue empleado en un curso de formación del LOTUS SYMPHONY.

Describe el paquete y sus reglas básicas, así como las tres funciones más frecuentes y las órdenes más comunes, sin olvidar las especiales que se dan sobre la hoja de trabajo. También enseña el modo de presentación de los datos en forma de gráficos y los programas de utilidades para crear archivos de datos.

Autor: Dick Waller.

Precio: 583 ptas.

Comercializa: Grupo Distribuidor Editorial.



# GESTION DE ERRORES (y II)

***Enchufemos de nuevo el ordenador. ¿Han cometido muchos errores desde nuestra última reunión? Seguro que no, pero aunque así haya sido ya estamos en disposición de tratar cada uno de ellos de una forma adecuada. O ¿no?***



Pero todavía no hemos terminado. En la mayoría de las ocasiones en las que se produce un error, el programador (en este caso usted mismo, sin ir más lejos) no conoce de antemano de qué tipo va a ser ni en la línea donde ocurrirá.

Puede suceder entonces que preparemos una rutina de error que no le trate correctamente y por tanto no consigamos con ella los resultados apetecidos. Imagine algo parecido al Programa 1.

## Programa uno

Antes de ejecutarlo, analice con detenimiento cada una de sus líneas. ¿Hay algún error en ellas? Y, en caso afirmativo, ¿somos capaces de corregirlo o evitarlo?

Generalmente, cuando el **Amstrad** detecta un error bien sintáctico o de cualquier otra

clase, la ejecución del programa queda detenida y en la pantalla aparece un mensaje, en inglés por supuesto, que nos informa de lo ocurrido. El ordenador queda dispuesto —o Ready— para que le podamos introducir cualquier comando. Vimos que posee el llamado «gestor» de errores que se encarga de hacerlo.

Pero podemos manipular esto a nuestro antojo consiguiendo que en vez de devolver el control al sistema operativo incorporado al micro, lo haga a una rutina que intente subsanar el problema encontrado o, si no es posible, al menos darnos una información más clara y detallada de lo que le ha sucedido a nuestro programa.

Y esto es lo que sucede con el que tenemos entre manos. El mérito se lo debemos a la línea 20:

20 ON ERROR GOTO 70

que interrumpe la ejecución del programa y la desvía hacia la línea 70, donde comienza una rutina que nos va a contar más cosas sobre lo que ha pasado.

El fallo que hemos cometido, y que detecta nuestro **Amstrad**, es un sencillo error sintáctico que se nos ha «deslizado» en la línea 50, al olvidárenos poner la T de la palabra PRINT.

El **Amstrad** se encuentra con ello y salta a la rutina de la línea 70 que nos dice lo ocurrido, pero en castellano, y mediante el:

RESUME 60

de la línea 110, volvemos a nuestro programa principal justamente al END que hace que termine su ejecución.





# Primeros PASOS

Ahora, manos a la obra y a corregirlo. Ponga la T que faltaba y haga «correr» de nuevo el programa. Va todo sobre ruedas, ¿no?

Imonige que le añadimos muchísimas líneas más. Bueno, no se asuste porque en nuestro caso sólo con una será suficiente para conseguir los fines que deseamos. Escriba:

55 LOCATE 15.12

Con esta línea lo que queremos hacer es situar el cursor de textos en la celda de coordenadas 15.12 (o columna 15 y fila 12). Pero es evidente que no lo vamos a lograr, ya que de nuevo hemos cometido un error sintáctico —a posta, claro— en esta nueva línea: separar los valores de las coordenadas por un punto en vez de una coma.

Vale, teniendo en cuenta que el programa saltará a la rutina de tratamiento de error, ejecútelo otra vez. ¿Qué ocurre ahora?

Como en las ocasiones anteriores, el **Amstrad** encuentra el error e intenta solucionarlo —recuerde la misión de ON ERROR GOTO...—, pero esta vez no lo consigue o lo hace a medias. Nos da un mensaje de que existe una línea que no está escrita correctamente, pero lo que no es cierto es el número de orden indicado en el mismo: el error no está ahora en la línea 50, sino en la 55. ¿De acuerdo?

Este resultado ha sido debido, como dijimos antes, a que no siempre vamos a conocer el lugar exacto en el que se va a producir el error, sino que generalmente surgirá donde menos le esperemos, excepto que lo hayamos preparado, como en este caso.

```
10 REM PROGRAMA I
20 ON ERROR GOTO 70
30 CLS
40 saludo$="HOLA"
50 PRIN saludo$
60 END
70 REM Rutina Tratamiento de Error
80 PRINT
90 PRINT"ERROR SINTACTICO EN LA LINEA 50"
100 PRINT
110 RESUME 60
```

Por eso no podemos escribir una rutina tan particular. Y, si es más general, ¿de qué nos sirve? Piense qué ocurriría si el mensaje de error fuera solamente:

## ERROR SINTACTICO

La información que nos aporta es bastante escasa. Si no conocemos ni la línea donde ha ocurrido nos será bastante más difícil corregirlo, sobre todo si el programa es más largo

y complicado que el del ejemplo y tenemos que ir examinándolo instrucción por instrucción. ¿Qué hacemos entonces?

De momento no ponernos nerviosos ni histéricos. El **Amstrad** nos sacará del apuro. Está provisto de una función que es capaz de darnos la información adicional que nos falta: el número de línea de la instrucción donde se produjo el error. Se trata de ERL.

Y simplemente hace eso, pero de momento ya es suficiente. Le proponemos ahora una especie de ejercicio. Teclee el Programa 2 y ejecútelo.

## Programa dos

¿Qué diferencia existe entre el resultado final obtenido y el que se produjo con el pro-

```
10 REM PROGRAMA 11
20 ON ERROR GOTO 70
30 CLS
40 saludo$="HOLA"
50 PRIN saludo$
60 END
70 REM Rutina Tratamiento de Error
80 PRINT
90 PRINT"ERROR SINTACTICO EN LA LINEA";ERL
100 PRINT
110 RESUME 60
```

grama anterior? Aparentemente es el mismo, pero el modo de llegar a él es diferente.

En ambos nos aparece un mensaje en la pantalla en el que el ordenador nos dice dónde hemos cometido el fallo: línea 50. Ahora bien, en el primero formábamos el texto directamente mientras que en este segundo empleamos para ello la nueva función ERL.

Corrija, como antes, la línea 50 y añada la 55. Esta vez sí que podemos observar claramente los efectos de ERL, ya que la información sí que es exacta. El **Amstrad** nos dice el número de la sentencia donde se ha producido el error. Y todo ha sido debido al cam-



bio que hemos introducido en la línea 90.

Así pues, ya tenemos un poco más localizado el error y por tanto será bastante más fácil corregirlo. Pero compliquemos un poco más las cosas para intentar no aburrirnos. Vamos a mezclar ahora varios diferentes a ver qué pasa.

El Programa 3 es una pequeña mezcla de lo que le acabamos de decir. Escríbalo tal como aparece el listado —errores incluidos— y analícelo con detenimiento antes de ejecutarlo.

### Programa tres

Tenemos, como antes, un error sintáctico en la línea 50, y además un RETURN inesperado en la 60. Decimos que es inesperado debido a que llegamos a él sin que se haya producido ninguna llamada a subrutina. ¿De acuerdo?

Y visto esto, ejecútelo. Observe que el que se va a encargar de tratar ahora cada uno de ellos es el gestor de errores del Basic, ya que, de momento, no hemos introducido ninguna línea donde indiquemos que se transfiera el control a la rutina preparada para ello, que por ahora está de «adorno».

```
10 REM PROGRAMA III
30 CLS
40 saludo$="HOLA"
50 PRIN saludo$
60 RETURN
70 END
80 REM Rutina Tratamiento de Error
90 PRINT
100 PRINT "ERROR SINTACTICO EN LA LINEA"; ERL
110 PRINT
120 RESUME NEXT
```

El Basic primero detectará el error sintáctico y sacará el mensaje:

Syntax error in 50

para devolver a continuación el control al sistema operativo que nos permitirá corregirlo. ¡Hágalo, pues!

Ponga de nuevo el Programa en marcha con RUN. Ahora, una vez eliminado el sintáctico, el ordenador encontrará un RETURN donde no debía estar y nos avisa de ello:

Unexpected RETURN in 60

Introduzcamos a continuación nuestro pro-

prio gestor de errores. Para ello añade al Programa 3 la siguiente línea:

20 ON ERROR GOTO 80

que será la encargada de interrumpir la ejecución del programa, cuando sea preciso, y desviarla hacia la rutina que nos dará una información concreta de lo que está sucediendo y que comienza en la línea indicada al final de la instrucción ON ERROR...

Demos marcha atrás y repitamos el proceso. Teclee RUN y observe qué es lo que ocurre.

El ordenador detecta el primer error y nos lo dice con número de sentencia incluida. Mediante él:

RESUME NEXT

de la línea 120, la ejecución continúa en la siguiente a la indicada por ERL y sigue hasta que se encuentra con el RETURN incorrecto.

Hay un nuevo error, así que volvemos a la rutina que los trata. Aparece un mensaje en la pantalla y el programa acaba porque RESUME NEXT hace que siga por la línea 70, donde se encuentra con END.

¿Qué sacamos en claro de todo esto? La verdad es que poca cosa, ya que sólo hemos llegado a conocer el número de las líneas donde se han producido anomalías. Pero éstas han sido debidas a causas bien diferentes, como ya sabemos.

Estamos echando en falta algo que nos permita distinguir qué es lo que ha pasado en cada caso para así ir directos al grano e intentar solucionar nuestro problema.

Pues bien, lo mismo que existe una función ERL que nos devuelve el número de la línea de la instrucción donde se ha producido un error, el **Amstrad** también posee otra que nos informará del número o tipo de error detectado. Es justamente lo que necesitamos.

Con la función ERR el ordenador nos dirá cuál de todos los tipos de error que encontrará en la lista de mensajes, incluida en el Manual del Usuario, es el que se ha producido.

Y si podemos disponer de ella, ¿por qué no utilizarla? Con el Programa 4 podemos ver una sencilla manera de tratar diferentes tipos de errores. Seguro que no tiene ningún problema para entenderlo.

### Programa cuatro

Es una variación del anterior, pero en el que ya hemos incluido la nueva función ERR en la rutina que va «gestionar» los diferentes errores.

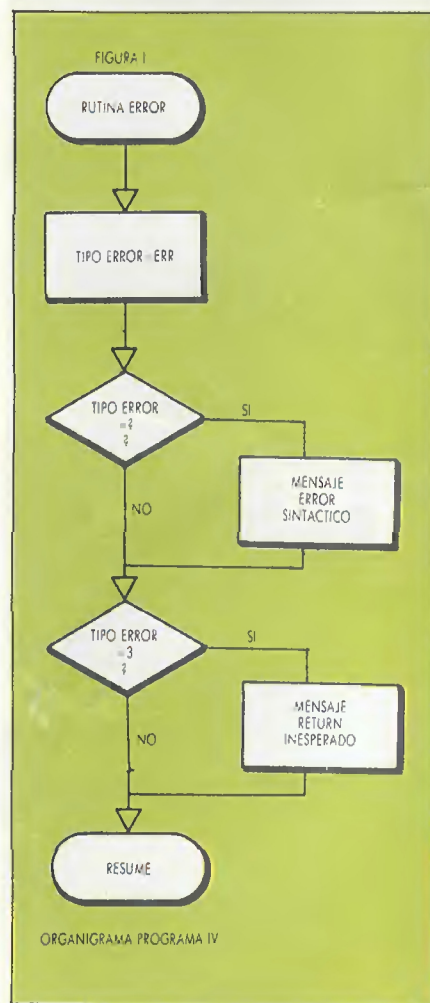
Lo primero que hace es pasar el valor que nos devuelve esta función a la variable «**numerror**» en la línea 90 y después analiza el contenido de la misma.

Si «**numerror**» vale 2 estamos ante un error sintáctico, tal como nos lo indica la tabla a la

que nos referimos anteriormente, y como tal lo tratamos.

Sin embargo, cuando el tipo de error, o el valor contenido en «**numerror**», sea el 3 tendremos que sacar un mensaje en el que digamos que en la línea 60 el ordenador ha encontrado un RETURN que no esperaba. Ojee de nuevo su Manual para comprobar que el 3 es el número de error que se ha producido.

Todo esto responde a un sencillo proceso lógico que podemos resumir con un organigrama que quizá nos haga ver las cosas un poco más claras. La figura 1 es la representación gráfica de lo que le acabamos de decir. Analícela y seguro que le ayudará a despejar alguna de sus dudas si es que las tuviera.



Hagamos una pequeña recapitulación de estas dos últimas ideas. Existen en el Basic del **Amstrad** dos funciones que nos servirán para encontrar una información adicional sobre los errores cometidos, informáticamente hablando. ERL nos da un número que corresponde al de la línea de la sentencia donde se ha detectado el último error, mientras que ERR nos devuelve también otro número, pero que ahora es el tipo del último error detectado y para conocer todos los que existen le remitimos al Manual.

Hay ocasiones en las que dentro de un programa no existe ningún error de los que pue-



den ser detectados por el Basic, pero en el que, sin embargo, podemos estar introduciendo datos, sin ser errores, quizá no sean correctos.

Nos explicamos. Imagine que tenemos que teclear un año de este siglo. Es evidente que su valor estará comprendido entre 1900 y 1999, ¿no?

La forma de detectarlo hasta ahora era mediante dos líneas de programa semejantes a:

```
100 INPUT «AÑO»; a
110 IF a < 1900 OR a > 1999
    THEN GOTO 100
```

Veámoslo ahora bajo otra perspectiva. Si contestamos con un año que no está dentro de este rango estamos cometiendo un error. En los datos, pero al fin y al cabo, es error. Y así vamos a tratarlo.

En nuestras múltiples ojeadas al Manual en este artículo habrá observado que están contemplados 32 tipos de error que podemos decir sean los clásicos del Basic. Al nuestro, como es nuevo, le daremos el número 33.

```
10 REM PROGRAMA IV
20 ON ERROR GOTO 80
30 CLS
40 saludo$="HOLA"
50 PRINT saludo$
60 RETURN
70 END
80 REM RUTINA TRATAMIENTO DE ERROR
90 numerror=ERR
100 PRINT
110 IF numerror=2 THEN PRINT"ERROR
    SINTACTICO EN LA LINEA";ERL
120 IF numerror=3 THEN PRINT"RETURN
    INESPERADO EN LA LINEA";ERL
130 PRINT
140 RESUME NEXT
```

¿Cómo hacer ahora que nuestro propio gestor de errores lo detecte y lo trate como si fuera uno de los contemplados por el Basic? Pues muy sencillo, incluyendo una línea en el programa tal como:

```
120 IF a < 1900 OR a > 1999
    THEN ERROR 33
```

¿Así de sencillo? Pues sí, así de sencillo. ERROR no es una palabra cualquiera sino que está reservada para una orden Basic. Y, ¿qué hace esta instrucción?

Lo primero es que activa nuestro «gestor» a medida y éste hace que el programa salte a la rutina de tratamiento de errores, si es que la hay, para intentar corregirlo. Además genera unos valores correctos con las funciones ERR y ERL que añaden más información al tema.

Podemos decir que con ERROR estamos fabricándonos una tabla de errores cuyos tipos están descritos por el mismo y por supuesto fuera del rango de los que puede detectar el Basic. El Programa 5 es una muestra de cómo podemos utilizar esta instrucción en la práctica.

## Programa cinco

Es un ejemplo muy sencillito en el que se nos pide introducir por pasos una fecha de este siglo, para continuar con lo que ya hemos comenzado.

Hemos catalogado tres nuevos errores de datos: el 33 lo asociamos a haber tecleado un día incorrecto, el 34 se producirá cuando el valor del mes no sea válido y con el 35 podemos decir lo mismo, pero para el año.

Observe que invocamos el error especificado en las líneas 50, 70 y 90, dependiendo en cada caso del fallo de datos producido y desde allí saltamos a la rutina de error.

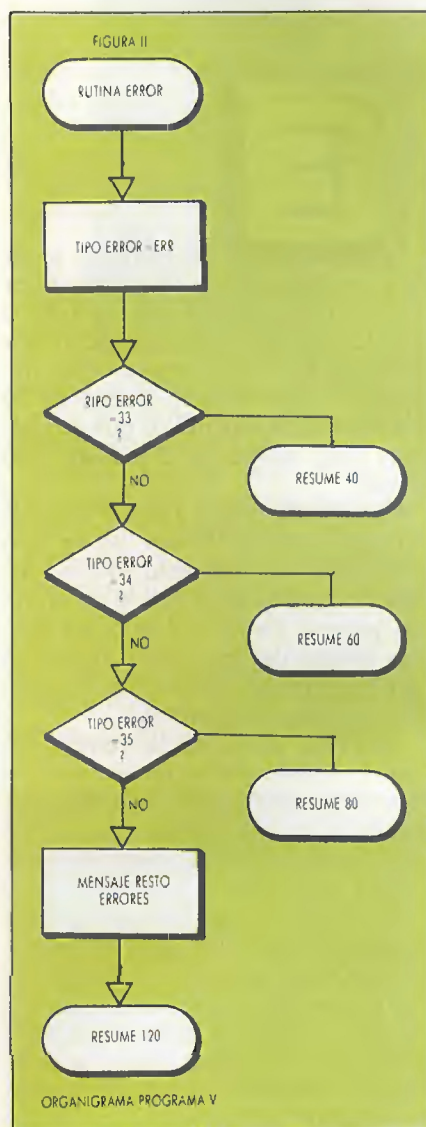
```
10 REM PROGRAMA V
20 ON ERROR GOTO 1000
30 CLS
40 INPUT"DIÁ: ";dia
50 IF dia<1 OR dia>31 THEN ERROR 33
60 INPUT"MES: ";mes
70 IF mes<1 OR mes>12 THEN ERROR 34
80 INPUT"AÑO: ";a
90 IF a<1900 OR a>1999 THEN ERROR 35
100 PRINT
110 PRINT"La fecha es";dia;"del";mes;"de";a
120 END
1000 REM TRATAMIENTO DE RUTINA DE ERROR
1010 numerror=ERR
1020 IF numerror=33 THEN RESUME 40
1030 IF numerror=34 THEN RESUME 60
1040 IF numerror=35 THEN RESUME 80
1050 PRINT"ERROR";numerror;"EN LA LINEA";ERL
1060 RESUME 120
```

En ella se hace un análisis del valor que nos devuelve ERR (o el tipo de error) y se actúa en consecuencia. También hemos contemplado el caso en el que se ha producido un error de un tipo diferente a los que nosotros hemos definido y la rutina lo tratará igualmente.

Le recomendamos que estudie la figura 2. En ella está reflejado el camino lógico que ha

# Primeros PASOS

seguido el **Amstrad** cuando ha encontrado un error. Primero lo analiza y después de informarnos de lo que pasa devuelve el control al punto del programa más conveniente en cada caso.



Recuerde que es muy útil emplear estas representaciones gráficas, u organigramas, antes de ponerse a codificar como un loco. Le clarificarán mucho las cosas y además le será muy sencillo transformar posteriormente a instrucciones Basic.

Y por el momento nada más. Esta vez le deseamos que procure cometer muchos errores para poder explorar a fondo todas las posibilidades de las que dispone su ordenador para tratarlos. Así que..., hasta pronto.



# GENCHAR

*En muchas ocasiones, nos hemos visto obligados a definir un nuevo juego de caracteres para que el programa que estamos realizando, quede con una presentación mucho más vistosa. Hemos tenido que ir componiendo uno a uno todos los caracteres que nos han interesado.*



llo significa tener que introducir en nuestro ordenador 8 bytes por cada carácter a definir, por lo que si se han definido, por ejemplo, las letras mayúsculas y minúsculas, éstas forman un total de 50 caracteres, lo que indica que nos hemos visto obligados a teclear ¡400 bytes!

A partir de ahora y con el programa que os presentamos, ya no será necesario efectuar esta penosa tarea, ya que nuestra rutina es capaz de generar cientos y cientos de nuevos juegos de caracteres originales que podrán ser introducidos en nuestros propios programas.

Además, presenta la ventaja de que la nueva definición se efectúa desde el carácter 32 hasta el carácter 255, lo cual significa poseer un juego completo de caracteres totalmente originales.

Por si fuera poco, dicho programa aún nos ofrece una ventaja adicional, y es que cada uno de los distintos juegos de caracteres se pueden obtener mediante comandos RSX, por lo que no existirá ningún problema a la hora de introducirlo en los programas que estemos confeccionando.

Por si fueran pocas las posibilidades que nos proporciona esta rutina, aún queda otra, que por sí sola nos será de una gran utilidad tanto si trabajamos con los nuevos caracteres como si trabajamos con los propios de **Amstrad**.

Esta última opción nos va a permitir imprimir en pantalla cualquier carácter en cualquier sentido. Es decir, podremos escribir de abajo a arriba,

o bien en sentido inverso al normal, todo ello con unos caracteres en los que se habrá producido una rotación de 90, 180 o 270 grados, según el sentido en que se esté imprimiendo.

Por si lo dicho hasta el momento os parece poco, debemos decir por último que la rutina capaz de producir todos estos efectos ocupa únicamente ¡496 bytes!, y se ubica a partir de la dirección hexadecimal &9500.

Vamos a describir a continuación cada uno de los nuevos comandos RSX capaces de generar los nuevos juegos de caracteres gráficos.

**ICARS0:** Coloca el juego de caracteres original de **Amstrad** y coloca dichos caracteres en las posiciones iniciales, sea cual fuere su sentido de giro en ese momento.

**ICARS1:** Produce caracteres a doble anchura de la normal.

**ICARS2:** Genera letras normales partidas por la mitad en forma vertical.

**ICARS3:** Se obtiene un juego de caracteres estrecho pero grueso.

**ICARS4:** También genera letras estrechas pero en este caso delgadas al mismo tiempo.

**ICARS5:** Caracteres partidos horizontalmente.

**ICARS6:** Se consigue un juego de caracteres algo más corto de lo normal (utilizable en especial para las letras mayúsculas).

**ICARS7:** Provoca un alargamiento en el juego de caracteres actual.



**ICARS8:** Realiza un desplazamiento escalonado hacia la derecha y de abajo a arriba del juego de caracteres actual, con lo que se obtienen letras inclinadas hacia la derecha.

**ICARS9:** Es idéntica a la anterior pero en este caso se produce el efecto hacia la izquierda.

**IGIRO:** Provoca una rotación de 90 grados en el actual juego de caracteres. Si se efec-





cuenta que la unión de dos juegos de caracteres, en la mayoría de los casos no es conmutativa, es decir no es lo mismo efectuar:

ICARS8: ICARS3

que ejecutar:

ICARS3: ICARS8

Realmente, no sabría decir cuántos nuevos tipos de caracteres se pueden conseguir a través de esta rutina, ya que me ha sido totalmente imposible poder realizar todas las combinaciones, pero en el programa demostración se muestran algunos ejemplos.

Algunas de las combinaciones binarias con las que se pueden obtener unos bonitos gráficos, son las siguientes:

ICARS4: ICARS8  
ICARS0: ICARS7  
ICARS7: ICARS8  
ICARS1: ICARS8

Aunque verdaderamente existen muchas más posibilidades de combinación.

Un ejemplo de combinación de tres tipos de caracteres, puede ser la siguiente:

ICARS4: ICARS9: ICARS1

Como podéis comprobar, los tipos de variaciones son muchísimos, únicamente basta con investigar cada una de las posibles combinaciones.

Del mismo modo se pueden obtener diferentes juegos de caracteres gráficos, uniendo cuatro, cinco o hasta los nueve tipos distintos de caracteres «base».

Dado que efectuando algunas de las combinaciones posibles, podemos obtener letras ilegibles, tenemos la posibilidad de efectuar:

ICARS0

con lo que obtendremos el juego original de caracteres que posee el **Amstrad**.

Del mismo modo para obtener cada uno de los tipos «base», deberemos ejecutar previamente el comando anterior.

Para grabar la presente rutina en disco o cassette, únicamente deberéis copiar el listado ensamblador que aparece a continuación, colocando como origen la dirección hexadeci-

## PROGRAMACION

mal &9500, y salvarlo como código objeto.

Aquellos que prefieran teclear el cargador Basic, una vez terminado de copiar, deberán ejecutar el programa, y en el caso de que aparezca algún mensaje de error, deberán revisar las líneas «DATA». Si el programa se ejecuta correctamente, se procederá a salvarlo en cinta o disco de la forma siguiente:

SAVE«CHARS»,B,&9500,500

con lo cual ya podrá ser utilizado como se desee.

Cuando queramos cargar el código máquina en memoria, deberemos ejecutar un programa Basic como el siguiente:

10 MEMORY &94FF  
20 LOAD«CHARS»,&9500  
30 CALL &9500

con lo cual se tendrá el programa en memoria y se habrán inicializado los nuevos comandos RSX.

A partir de este momento podremos empezar a escribir nuestro programa con la posibilidad de utilización de los once nuevos comandos.

No resta nada más por decir, ya que vosotros mismos podréis ver la gran potencia de esta rutina en código máquina. A disfrutar y a ver quién consigue mayor número de juegos de caracteres distintos.

## LISTADO DEENSAMBLADO

10	ORG	#9500
20	LD	DE, 32
30	LD	HL, BUFLET
40	CALL	#BBAB
50	CALL	STORE
60	LD	BC, TABLA
70	LD	HL, ESPACE
80	JP	#BCD1
90	TABLA:	DEFW NAME
100	JP	LETRA
110	JP	CHANGE
120	JP	CHANG1
130	JP	CHANG2
140	JP	CHANG3
150	JP	CHANG4
160	JP	CHANG5
170	JP	CHANG6
180	JP	CHANG7
190	JP	CHANG8

túa dos veces se obtiene una rotación de 180 grados, y de 270 si se efectúa por tercera vez. En esta situación, si se vuelve a ejecutar se obtienen los caracteres con el sentido de giro original.

Así pues, existen nueve juegos de caracteres «base» totalmente diferentes e independientes unos de otros, por lo que efectuando combinaciones entre ellos, se podrán conseguir numerosísimos nuevos juegos de caracteres, más aún si tenemos en



200		JP	GIRO	900	LD	A,B	1600	CCF			
210	NAME:	DEFM	"CARS"	910	OR	C	1610	RR	(IX+2)		
220		DEFB	"0"+#80	920	JR	NZ,BUCL2	1620	SCF			
230		DEFM	"CARS"	930	RET		1630	CCF			
240		DEFB	"1"+#80	940	CHANG3:	LD	HL,BUFLET	1640	RR	(IX+1)	
250		DEFM	"CARS"	950		LD	BC,1800	1650	SCF		
260		DEFB	"2"+#80	960	BUCL3:	LD	A,(HL)	1660	CCF		
270		DEFM	"CARS"	970		RR	(HL)	1670	RR	(IX+1)	
280		DEFB	"3"+#80	980		RR	(HL)	1680	SCF		
290		DEFM	"CARS"	990		LD	D,A	1690	CCF		
300		DEFB	"4"+#80	1000		RL	D	1700	RR	(IX+0)	
310		DEFM	"CARS"	1010		AND	D	1710	SCF		
320		DEFB	"5"+#80	1020		LD	(HL),A	1720	CCF		
330		DEFM	"CARS"	1030		INC	HL	1730	RR	(IX+0)	
340		DEFB	"6"+#80	1040		DEC	BC	1740	LD	D,0	
350		DEFM	"CARS"	1050		LD	A,B	1750	LD	E,8	
360		DEFB	"7"+#80	1060		OR	C	1760	ADD	IX,DE	
370		DEFM	"CARS"	1070		JR	NZ,BUCL3	1770	DJNZ	BUCL7	
380		DEFB	"8"+#80	1080		RET		1780	RET		
390		DEFM	"CARS"	1090	CHANG4:	LD	HL,BUFLET	1790	CHANG8:	LD	IX,BUFLET
400		DEFB	"9"+#80	1100		LD	B,223	1800		LD	B,223
410		DEFM	"GIR"	1110	BUCL4:	INC	HL	1810	BUCL8:	SCF	
420		DEFB	"0"+#80	1120		INC	HL	1820		CCF	
430		DEFB	0	1130		INC	HL	1830		RR	(IX+7)
440	ESPACE:	DEFS	4	1140		INC	HL	1840		SCF	
450	;			1150		INC	HL	1850		CCF	
460	;			1160		LD	(HL),0	1860		RR	(IX+7)
470	CHANGE:	LD	HL,BUFLET	1170		INC	HL	1870		SCF	
480		LD	BC,1800	1180		INC	HL	1880		CCF	
490	BUCL:	LD	A,(HL)	1190		INC	HL	1890		RR	(IX+6)
500		SCF		1200		DJNZ	BUCL4	1900		SCF	
510		CCF		1210		RET		1910		CCF	
520		RR	(HL)	1220	CHANG5:	LD	IX,BUFLET	1920		RR	(IX+6)
530		OR	(HL)	1230		LD	B,223	1930		SCF	
540		LD	(HL),A	1240	BUCL5:	LD	D,(IX+0)	1940		CCF	
550		INC	HL	1250		LD	C,(IX+1)	1950		RR	(IX+5)
560		DEC	BC	1260		LD	(IX+1),D	1960		SCF	
570		LD	A,B	1270		LD	(IX+2),C	1970		CCF	
580		OR	C	1280		LD	(IX+0),0	1980		RR	(IX+4)
590		JR	NZ,BUCL	1290		LD	DE,8	1990		SCF	
600		RET		1300		ADD	IX,DE	2000		CCF	
610	CHANG1:	LD	HL,BUFLET	1310		DJNZ	BUCL5	2010		RL	(IX+1)
620		LD	BC,1800	1320		RET		2020		SCF	
630	BUCL1:	RES	4,(HL)	1330	CHANG6:	LD	IX,BUFLET	2030		CCF	
640		INC	HL	1340		LD	B,223	2040		RL	(IX+0)
650		DEC	BC	1350	BUCL6:	LD	D,(IX+3)	2050		LD	D,0
660		LD	A,B	1360		LD	E,(IX+4)	2060		LD	E,8
670		OR	C	1370		LD	H,(IX+5)	2070		ADD	IX,DE
680		JR	NZ,BUCL1	1380		LD	L,(IX+6)	2080		DJNZ	BUCL8
690		RET		1390		LD	(IX+4),D	2090		RET	
700	CHANG2:	LD	HL,BUFLET	1400		LD	(IX+5),E	2100	STORE:	LD	HL,BUFLET
710		LD	BC,1800	1410		LD	(IX+6),H	2110		LD	DE,STORLE
720	BUCL2:	LD	A,(HL)	1420		LD	(IX+7),L	2120		LD	BC,1800
730		LD	(HL),0	1430		LD	D,0	2130		LDIR	
740		RRA		1440		LD	E,8	2140		RET	
750		RRA		1450		ADD	IX,DE	2150	LETRA:	LD	HL,STORLE
760		RR	(HL)	1460		DJNZ	BUCL6	2160		LD	DE,BUFLET
770		RRA		1470		RET		2170		LD	BC,1800
780		RR	(HL)	1480	CHANG7:	LD	IX,BUFLET	2180		LDIR	
790		RRA		1490		LD	B,223	2190		RET	
800		RR	(HL)	1500	BUCL7:	SCF		2200	GIRO:	LD	DE,BUFLET
810		RRA		1510		CCF		2210		LD	B,223
820		RRA		1520		RL	(IX+7)	2220	BUCL9:	PUSH	BC
830		RR	(HL)	1530		SCF		2230		LD	HL,BUFFER
840		RRA		1540		CCF		2240		LD	B,8
850		RR	(HL)	1550		RL	(IX+6)	2250	NUD4:	PUSH	BC
860		RRA		1560		SCF		2260		PUSH	HL
870		RR	(HL)	1570		CCF		2270		LD	B,8
880		INC	HL	1580		RR	(IX+3)	2280		LD	A,(DE)
890		DEC	BC	1590		SCF		2290	NUD3:	RRA	



```

2300      RL      (HL)
2310      INC     HL
2320      DJNZ    NUD3
2330      POP     HL
2340      INC     DE
2350      POP     BC
2360      DJNZ    NUD4
2370      PUSH    DE
2380      EX      DE,HL
2390      LD      DE,8
2400      SCF
2410      CCF
2420      SBC     HL,DE
2430      EX      DE,HL
2440      LD      HL,BUFFER
2450      LD      BC,8
2460      LDIR
2470      POP     DE
2480      POP     BC
2490      DJNZ    BUCL9
2500      RET
2510 BUFFER: DEFS 8
2520 BUFLET: DEFS 1800
2530 STORLE: DEFS 1800

```

### TABLA DE ETIQUETAS

BUCL2	959C	BUCL3	95C0
BUCL5	95E9	BUCL6	9607
BUCL8	966D	BUCL9	96C3
BUFFER	96F0	BUFLET	96F8
CHANG2	9596	CHANG3	95BA
CHANG5	95E3	CHANG6	9601
CHANG8	9667	CHANGE	9573
GIRO	96BE	LETRA	96B2
NUD3	96CE	NUD4	96C9
STORLE	9E00	TABLA	9515

### PROGRAMA DEMOSTRACION

```

10 MEMORY &94FF
20 LOAD "chars",&9500
30 CALL &9500
40 MODE 1:INK 0,0:INK 1,1:INK 2,0:
INK 3,15:PAPER 0:BORDER 0
50 a$="MICROHOBBY AMSTRAD"
60 :CARS0
70 GOSUB 390
80 :CARS1
90 GOSUB 390
100 :CARS2
110 GOSUB 390
120 :CARS3
130 GOSUB 390
140 :CARS4
150 GOSUB 390
160 :CARS5
170 GOSUB 390
180 :CARS6
190 GOSUB 390
200 :CARS7
210 GOSUB 390

```

```

220 :CARS8:GOSUB 390
230 :CARS9:GOSUB 390
240 :CARS2:::CARS8:GOSUB 390
250 :CARS2:::CARS7:::CARS8:GOSUB 390
260 :CARS4:::CARS8:GOSUB 390
270 :CARS1:::CARS7:GOSUB 390
280 :CARS1:::CARS4:::CARS3
290 GOSUB 390
300 :CARS1:::CARS2
310 GOSUB 390
320 :CARS1:::CARS6
330 GOSUB 390
340 :CARS1:::CARS5
350 GOSUB 390
360 :CARS3:::CARS1
370 GOSUB 390
380 GOTO 60
390 :GIRO:PEN 1
400 FOR N=1 TO 18
410 LOCATE 2,22-N:PRINT MID$(A$,N,1)
420 NEXT
430 :GIRO
440 FOR N=1 TO 18
450 LOCATE 29-N,23:PRINT MID$(A$,N,1)
460 NEXT
470 :GIRO
480 FOR N=1 TO 18
490 LOCATE 38,N+3:PRINT MID$(A$,N,1)
500 NEXT
510 :GIRO
520 FOR N=1 TO 18
530 LOCATE N+10,2:PRINT MID$(A$,N,1)
540 NEXT
550 PEN 2:LOCATE 13,7:PRINT "PULSA U
NA TECLA":PEN 3:LOCATE 9,15:PRINT "
PARA CAMBIAR CARACTERES"
560 WHILE INKEY="" :WEND
570 :CARS0
580 RETURN

```

### PROGRAMA CARGADOR

```

10 REM *PROGRAMA CARGADOR*
20 FOR N=&9500 TO &96F8
30 READ A:SUMA=SUMA+A
40 POKE N,A
41 NEXT
50 IF SUMA<>52132 THEN PRINT "ERROR EN DATAS"
70 DATA 17,32,0,33,248,150,205
80 DATA 171,187,285,166,158,1,21
90 DATA 149,33,111,149,195,209,188
100 DATA 56,149,195,178,150,195,115
110 DATA 149,195,135,149,195,150,149
120 DATA 195,186,149,195,209,149,195
130 DATA 227,149,195,1,150,195,40
140 DATA 150,195,183,150,195,190,150
150 DATA 67,65,82,83,176,67,65
160 DATA 82,83,177,67,65,82,83
170 DATA 178,67,65,82,83,179,67
180 DATA 65,82,83,180,67,65,82
190 DATA 83,181,67,65,82,83,182
200 DATA 67,65,82,83,183,67,65

```

## PROGRAMACCION

```

210 DATA 82,83,184,67,65,82,83
220 DATA 185,71,73,82,207,0,0
230 DATA 0,0,0,33,248,150,1
240 DATA 8,7,126,55,63,203,38
250 DATA 182,119,35,11,120,177,32
260 DATA 243,201,33,248,150,1,8
270 DATA 7,203,166,35,11,120,177
280 DATA 32,248,201,33,248,150,1
290 DATA 8,7,126,54,0,31,31
300 DATA 203,30,31,203,30,31,203
310 DATA 30,31,31,203,30,31,203
320 DATA 30,31,203,30,35,11,120
330 DATA 177,32,227,201,33,248,150
340 DATA 1,8,7,126,203,30,203
350 DATA 38,87,203,18,162,119,35
360 DATA 11,120,177,32,248,201,33
370 DATA 248,150,6,223,35,35,35
380 DATA 35,35,54,0,35,35,35
390 DATA 16,244,201,221,33,248,150
400 DATA 6,223,221,86,0,221,78
410 DATA 1,221,114,1,221,113,2
420 DATA 221,54,0,0,17,8,0
430 DATA 221,25,16,233,201,221,33
440 DATA 248,150,6,223,221,86,3
450 DATA 221,94,4,221,102,5,221
460 DATA 110,6,221,114,4,221,115
470 DATA 5,221,116,6,221,117,7
480 DATA 22,0,30,8,221,25,16
490 DATA 224,201,221,33,248,150,6
500 DATA 223,55,63,221,203,7,22
510 DATA 55,63,221,203,6,22,55
520 DATA 63,221,203,3,30,55,63
530 DATA 221,203,2,30,55,63,221
540 DATA 203,1,30,55,63,221,203
550 DATA 1,30,55,63,221,203,0
560 DATA 30,55,63,221,203,0,30
570 DATA 22,0,30,8,221,25,16
580 DATA 200,201,221,33,248,150,6
590 DATA 223,55,63,221,203,7,30
600 DATA 55,63,221,203,7,30,55
610 DATA 63,221,203,6,30,55,63
620 DATA 221,203,6,30,55,63,221
630 DATA 203,5,30,55,63,221,203
640 DATA 4,30,55,63,221,203,1
650 DATA 22,55,63,221,203,0,22
660 DATA 22,0,30,8,221,25,16
670 DATA 200,201,33,248,150,17,0
680 DATA 158,1,8,7,237,176,201
690 DATA 33,0,150,17,248,150,1
700 DATA 8,7,237,176,201,17,248
710 DATA 150,6,223,197,33,248,150
720 DATA 6,0,197,229,6,8,26
730 DATA 31,203,22,35,16,250,225
740 DATA 19,193,16,240,213,235,17
750 DATA 8,0,55,63,237,82,235
760 DATA 33,248,150,1,8,0,237
770 DATA 176,209,193,16,212,201,0

```



**P** ara que tus dedos  
 no resienten el trabajo duro, M.H. AMS-  
 TRAD lo hace por ti. Todos los teclados que incluyen  
 este logotipo se encuentran a tu disposición en un cas-  
 sette mensual, solicitaselo.



# GREEN BERET

*Dentro de la base de misiles, una máquina entrenada para matar se abre paso ante el incesante ataque del personal de vigilancia. Desde luego, los juegos en los que un hombre solo se enfrenta a todo un ejército están de moda.*

**T**

cdo empezó con la anterior película de Rocky (Silvester Stallone), que por una vez y siguiendo las huellas de acorralado dejó de ser el boxeador que más galletas se lleva por combate, para convertirse en Rambo.

Rambo es un guerrillero que se introduce en un campamento Vietcong, para liberar a sus compañeros presos en la zona de alta seguridad.

Tras Rambo, apareció un juego para máquinas de billares basado en la misma temática, nos referimos al renombrado Comando de la casa Capcom.

Después del éxito de la película de Stallone, ésta tuvo su adaptación a juego para ordenador, pudiendo los poseedores de un **Amstrad** disponer de una versión para nuestro ordenador.



Compatible: CPC1664, CPC1664 y CPC16128.



Camino distinto pero con el mismo resultado fue el recorrido por el Comando. En este caso fue el juego de ordenador el que dio lugar a la película, apareciendo más tarde la versión para **Amstrad**.

Como dos juegos sobre el mismo tema son poca cosa, hemos tenido noticia de la existencia de otro programa similar, estamos hablando del mercenario.

Y por si esto fuera poco, y para que reine la abundancia, otro programa con la misma temática ha sido realizado para los amantes de la acción.

Su antecesor, un juego de billares de última moda, basado en una acción trepidante, y en el que como si de Swachernager se tratara, la máquina de matar liquida uno tras otro lo que se le ponga por delante.

Estamos hablando del **GREEN BERET**, nada más y nada menos que el boina verde, desde luego con este título algo gordo tenía que esperarnos.



El producto viene avalado por una casa de larga tradición en la producción de software; los inefables **KONAMI**.

La renombrada casa japonesa, es una experta en la fabricación de software para máquinas de billares. Todos recordamos su gran bum en este campo con los famosos **HYPER SPORTS**, más conocidos vulgarmente como las olimpiadas.

Hasta entonces, nadie había tratado la figura humana como **KONAMI**, los gráficos utilizados para los atletas y el movimiento del que estaban dotados, hace de **HYPER SPORTS**, un auténtico clásico de los megarcades.

La vena deportiva continuó con **HIPER MEDALLIST**, después llegaron las adaptaciones de los mismos para ordenadores domésticos, de las cuales hemos podido disfrutar en nuestros monitores.

Uno de los últimos productos de **KONAMI** para máquinas de billares, es el **GREEN BERET**, que ha si-





do adaptado en su versión para ordenador doméstico por la casa IMAGINE.

Los que hayan podido contemplar la versión de los billares, habrán apreciado las excelencias del juego a que nos referimos, en el que los insuperables gráficos y la imparable marcha del boina verde, nos hacen pegarnos a la pantalla.

Pues por una vez y sin que sirva de precedente, nos encontramos con una versión para el **Amstrad** del famoso juego, que no desmere-

ce en nada al juego original para megaarcades.

Por arte del software, nos hemos introducido en el cuerpo de un boina verde, el cual pretende una base soviética de alta seguridad.

Nuestra arriesgada marcha, comienza en las inmediaciones de un interminable puente de ferrocarril, que separa la base del resto de las instalaciones al otro lado del río.

Armado solamente con un cuchillo de monte, el boina verde se abre paso entre los primeros defensores que nos salen al paso; algunos de los cuales van armados con ametralladoras, morteros y armas pesadas que nos pueden destruir si no actuamos con rapidez.

Para conseguir armas más sofisticadas que un simple cuchillo de monte, hemos de eliminar al comandante que aparece ante nosotros en contadas ocasiones, de las cuales no podemos dejar pasar una.

Con su aniquilación, podemos conseguir tres tipos de armas; lanzallamas, lanzacohetes y granadas.

Cuando nos hacemos con una de estas armas, nuestra marcha se facilita en gran medida; primero golpeando con ellas a los defensores que nos acosan y disparándolas, con lo cual causamos gran número de bajas.

Cada una de estas armas dispone de tres disparos antes de quedar inutilizada, por lo que debemos reservarlas para las zonas en que las oleadas de atacantes son más intensas (generalmente la final de cada fase, cuando suena la alarma).

Con nuestra inquebrantable moral, hemos de atravesar una tras otra las distintas zonas de la base, hasta llegar al campo de prisioneros, que es nuestro objetivo final.

Sin lugar a dudas, nos encontramos ante el mejor de los juegos que abordan la temática iniciada con Rambo y seguida por el Comando.

En **GREEN BERET**, la acción es trepidante, somos acosados por todos lados, nuestro hombre dispone de una amplia gama de movimientos; puede subir, bajar, saltar hacia delante, hacia detrás, tumbarse y correr.

La realización de las distintas pantallas, está acompañada de unos decorados muy efectivos, que nos trasladan desde el puente de hierro hasta la base de misiles; entre camiones portamisiles, cañones de largo alcance y robustas estructuras metálicas.

El marco utilizado para la aven-

## Mister JOYSTICK



tura, y su puesta en escena, no pueden ser más acertados, las pantallas están realizadas a todo color y con una técnica de dibujo muy realista.

El tratamiento de los personajes, y su movimiento, son dignos de mención; en pantalla nos encontramos con soldados de cinco centímetros de altura, que se mueven con agilidad y van derechos al blanco.

Es de destacar el movimiento conseguido, porque además de la gama de movimientos del boina verde, hay ocasiones que nos encontramos en la misma pantalla con seis o siete soldados moviéndose simultáneamente, sin ralentizar la acción del programa.

Pero el cuidado puesto en la parte visual del programa, tiene su contrapeso en el sonido, este es un poco machacón y reiterativo, con una música poco imaginativa.

Un programa en fin, que satisfará las exigencias de los más expertos viciados de la acción, y amantes de los héroes solitarios que son capaces de aniquilar a todo un ejército.





# MICRO

***MICRO es un programa de gran interés para aquellos dedicados a la programación en código máquina. Su misión fundamental es controlar el microprocesador y concretamente, el contenido de los registros del mismo. Lo más interesante de este programa es la facilidad con que se puede manipular la CPU, permitiendo desde BASIC cargar los registros del Z80 y acceder a una rutina determinada, observando en todo momento los contenidos de los registros del micro.***



Los registros que controla este programa son los siguientes: A, B, C, D, E, F, H, L, IX, e IY; también controla, como es natural, los registros BC, DE y HL.

No se han incluido en él los registros SP, PC, I y R, pues su incorrecta utilización puede bloquear el sistema.

El programa se encuentra en su totalidad en BASIC, ahora bien, es por así decirlo, un pequeño generador de rutinas cortas en código máquina que se sitúan en una zona previamente señalada por el usuario.

## Cómo utilizar el programa MICRO

Bien, para utilizar el programa MICRO, lo cargaremos, y tras ponerlo en funcionamiento, responderemos a las preguntas que formule.

Habrà que indicar inicialmente dónde queremos que MICRO introduzca el pequeño programa que tiene que crear, por lo tanto, introduciremos la dirección de una zona libre de memoria.

A continuación introduciremos la rutina a la que queremos saltar (necesariamente debemos querer saltar a alguna rutina, pues si no fuese así, ¿para qué ver cuál es el contenido de los registros?

Una vez introducida la dirección de esta rutina, aparece en pantalla la relación de los registros que podemos cargar inicialmente, en el caso de que el usuario quiera modificarlos; en caso contrario simplemente con pulsar RETURN conseguiremos que ningún registro adquiera un nuevo valor. Si el usuario desea modificar algunos de los registros, tendrá que indicar previamente cuáles son; esto se consigue pulsando la letra de orden que aparece junto a ellos, de esta forma si queremos modificar el acumulador pulsaremos la a. Una vez indicadas todas las letras de orden, pulsaremos RETURN y ya sólo tendremos que introducir los valores con los que queremos cargarlos. Las entradas están controladas y no podrás cargar un registro de 8 bits con un dato de 16 bits.

Una vez que hemos acabado con la opción anterior podremos ver ya los contenidos de los registros según los valores asignados, pero previamente habremos de indicar si queremos que imprima por impresora tales contenidos.

En un principio, MICRO muestra los valores de los registros que el usuario ha introducido, y una vez visualizados, salta a la rutina indicada por él anteriormente, presentando por último los contenidos de los registros tras haber retornado de ella.

Si queremos saltar a otra rutina o repetir lo mismo con otros valores, pulsaremos una tecla cualquiera.



# Utilidades del PROGRAMADOR

## MODULOS DE MICRO

- 110-150** Inicialización de pantalla e introducción de la zona de operaciones y la rutina a la que se va a saltar.
- 200-240** Introducción de las letras de orden de los registros a cargar.
- 250-330** Introducción de los nuevos valores de los registros y formación de una cadena alfanumérica que contiene el código máquina que más tarde se POKEARA.
- 340-380** Se sigue generando la cadena alfanumérica b\$ que contiene el código máquina.
- 390-400** Selección de impresora o pantalla e introducción del código máquina en la zona de operaciones.
- 410-490** Visualización de los contenidos del MICRO antes y después de acceder a la rutina.
- 510-680** Rutina a la que se salta desde 410-490 para visualizar los contenidos.

## VARIABLES MAS IMPORTANTES

- \* b\$** Contiene el código máquina que se va generando según las necesidades del usuario.
- \* reg** Dirección a partir de la cual se guardan los contenidos de los registros para ser leídos por MICRO.
- \* flag\$** Almacena los 8 bits del registro F.
- \* ad** Contiene la dirección donde opera MICRO.
- i** Rutina a la que quiere saltar el usuario.

**Nota:** si deseas ver el código máquina creado por MICRO puedes teclear la siguiente sentencia:  
for i=ad to reg-1: ? hex\$(peek(i), 2): next.

## Estructura interna de MICRO

MICRO presenta una estructura un tanto compleja, pero se puede com-

prender fácilmente. La complejidad deriva de la forma como MICRO crea el SUBPROGRAMA en código máquina necesario para controlar la CPU. Además, esto se ve afectado también por la capacidad que tiene MICRO para situar esta rutina en M.C. en cualquier zona de memoria.

```

10 '*****
20 '** M I C R O '**
30 '** **
40 '** 8 de Julio de 1986 '**
50 '** ----- '**
60 '** Por G. LOPEZ S. '**
70 '** ----- '**
80 '*****
90 '
100 '
110 MODE 2:PRINT STRING$(80,154);:P
RINT" Julio 1986."SPC(17)CHR$(24)
" M I C R O "CHR$(24)SPC(17)"
Por G. Lo"CHR$(8)CHR$(22)"1"CHR$(16
1)CHR$(22)"0pez":PRINT STRING$(80,1
54);
120 WINDOW 1,88,4,25
130 LOCATE 1,22:PRINT CHR$(174)CHR$
(22)"1Do"CHR$(8)CHR$(161)"nde puedo
operar?"CHR$(22)"0";:INPUT " ",ad
140 IF ad<48 AND ad>-1 THEN CLS:GO
TO 130
150 CLS:LOCATE 1,22:PRINT CHR$(174)
CHR$(22)"1Do"CHR$(8)CHR$(161)"nde q
uieres saltar?"CHR$(22)"0";:INPUT "
",j
160 CLS
170 DIM a(11)
180 DATA A,B,C,D,E,H,L,BC,DE,HL,IX,
IY
190 DATA 3E,06,0E,16,1E,26,2E,01,11
,21,DD,FD
200 LOCATE 1,3:PRINT "INDICA LOS RE
GISTROS A MODIFICAR"
210 FOR i=0 TO 11:LOCATE 3,5+i:READ
d$:PRINT CHR$(97+i)"- "d$:NEXT
220 CLEAR INPUT: WHILE INKEY(6)=-1
AND INKEY(18)=-1
230 k$=INKEY$:IF k$="" THEN 240 ELS
E k$=LOWER$(k$):a=ASC(k$):IF a>96 A

```



```

ND a<109 THEN a=a-97:a(a)=1:LOCATE
3,5+a:PRINT CHR$(24)k$CHR$(24)
240 WEND
250 RESTORE 190:FOR i=0 TO 11
260 READ d$:IF a(i)=0 THEN 330
270 WHILE NOT INKEY$="":WEND
280 LOCATE 12,5+i:INPUT "Cargado a
",a
290 IF i>6 THEN i=2 ELSE i=1
300 IF (a>255 OR a<0) AND i=1 THEN
LOCATE 12,5+i:PRINT SPC(68):GOTO 28
0
310 IF i>9 THEN d$=d$+"21"
320 a$=HEX$(a,4):b$=b$+d$+RIGHT$(a$,
2):IF i=2 THEN b$=b$+LEFT$(a$,2)
330 NEXT
340 d=ad+LEN(b$)/2+30:op=d:j$=HEX$(
d,4):b$=b$+"CD"+RIGHT$(j$,2)+LEFT$(
j$,2)
350 d=d+3:reg=d:d$=HEX$(d,4):b$=b$+
"E5F5E122"+RIGHT$(d$,2)+LEFT$(d$,2)
+"E122":d=d+2:d$=HEX$(d,4):b$=b$+RI
GHT$(d$,2)+LEFT$(d$,2)
360 DATA ED43,ED53,DD22,FD22
370 FOR i=1 TO 4:d=d+2:READ k$:d$=H
EX$(d,4):b$=b$+k$+RIGHT$(d$,2)+LEFT
$(d$,2):NEXT
380 b$=b$+"C9C3":j$=HEX$(j,4):b$=b$
+RIGHT$(j$,2)+LEFT$(j$,2)
390 LOCATE 1,22:PRINT CHR$(174)"Sal
ida por impresora (S/N)?":INPUT "
",k$:k$=UPPER$(k$):IF k$="S" THEN p
rt=1 ELSE IF k$="N" THEN prt=0 ELSE
LOCATE 30,22:PRINT SPC(50):GOTO 3
90
400 o=ad:FOR i=1 TO LEN(b$) STEP 2:
a=VAL("&"+MID$(b$,i,2)):POKE o,a:o=
o+1:NEXT
410 CLS
420 a$=" SITUACION INICIAL DE LA CP
U ":PRINT CHR$(24)a$CHR$(24):IF prt
=1 THEN PRINT #8,a$:PRINT #8,STRING
$(29,"-")
430 PRINT:IF prt=1 THEN PRINT #8
440 POKE op,&C9:CALL ad:GOSUB 500
450 PRINT:IF prt=1 THEN PRINT #8

460 a$=" SITUACION FINAL DE LA CPU
":PRINT CHR$(24)a$CHR$(24): IF prt=
1 THEN PRINT #8,a$:PRINT #8,STRING$
(27,"-")
470 POKE op,&C3:CALL ad:PRINT: IF p
rt=1 THEN PRINT #8
480 GOSUB 500
490 LOCATE 32,22:PRINT CHR$(24)"PUL
SA TECLA"CHR$(24):WHILE INKEY$="":

```

```

WEND:RUN
500 flag$=BIN$(PEEK(reg),8)
510 a=PEEK(reg+1):a$="ACUMULADOR
":GOSUB 670
520 a$="BIT 0:C "+MID$(flag$,
8,1):PRINT a$:IF prt=1 THEN PRINT #
8,a$
530 a=PEEK(reg+5):a$="REGISTRO B
":GOSUB 670
540 a$="BIT 1:N "+MID$(flag$,
7,1):PRINT a$:IF prt=1 THEN PRINT #

```

```

8,a$
630 a=PEEK(reg+2):a$="REGISTRO L
":GOSUB 670
640 a=PEEK(reg+8)+PEEK(reg+9)*256:1
=LEN(STR$(a))-1:a$="REGISTRO IX "
+HEX$(a,4)+" "+RIGHT$("0000"+RIGHT$(
STR$(a),1),5)+STRING$(19,32):PRINT
a$:IF prt=1 THEN PRINT #8:PRINT #
8,a$;
650 a=PEEK(reg+10)+PEEK(reg+11)*256
:l=LEN(STR$(a))-1:a$="REGISTRO IY

```



```

8,a$
550 a=PEEK(reg+4):a$="REGISTRO C
":GOSUB 670
560 a$="BIT 2:P/V "+MID$(flag$,
6,1):PRINT a$:IF prt=1 THEN PRINT #
8,a$
570 a=PEEK(reg+7):a$="REGISTRO D
":GOSUB 670
580 a$="BIT 4:H "+MID$(flag$,
4,1):PRINT a$:IF prt=1 THEN PRINT #
8,a$
590 a=PEEK(reg+6):a$="REGISTRO E
":GOSUB 670
600 a$="BIT 6:Z "+MID$(flag$,
2,1):PRINT a$:IF prt=1 THEN PRINT #
8,a$
610 a=PEEK(reg+3):a$="REGISTRO H
":GOSUB 670
620 a$="BIT 7:S "+MID$(flag$,
1,1):PRINT a$:IF prt=1 THEN PRINT #

```

```

"+HEX$(a,4)+" "+RIGHT$("0000"+RIGH
T$(STR$(a),1),5):PRINT a$:IF prt=1
THEN PRINT #8,a$
660 RETURN
670 l=LEN(STR$(a))-1:a$=a$+HEX$(a,2
)+" "+RIGHT$("00"+RIGHT$(STR$(a),
1),3)+" "+BIN$(a,8)+STRING$(10,32
):PRINT a$:IF prt=1 THEN PRINT #8,
a$;
680 RETURN

```



**P** ara que tus dedos  
 no realicen el trabajo duro, M.H. AMS  
 TRAD lo hace por ti. Todos los listados que incluyen  
 este logotipo se encuentran a tu disposición en un cas-  
 sette mensual, solicitálos.



# CONDUCE TU AMSTRAD 464 A 512 K

... Y AGUANTA  
EL VERTIGO



## VORTEX SP-512 — Prodigiosa expansión de memoria.

Esta es la Placa VORTEX SP-512, capaz por sí sola de transformar tu ya conocido y dominado AMSTRAD 464 en un prodigio: 512 K de memoria, que multiplican hasta el vértigo las posibilidades de tu ordenador, gracias al sistema operativo VORTEX V-DOS incluido en la ROM.

Acude con tu AMSTRAD 464 a uno de los Servicios especializados VORTEX.

Una sencilla operación y la Placa VORTEX queda instalada. Se ha realizado la transformación

que te lleva a una nueva frontera. Pasa a "RAM EXPANSION". ¿Estás preparado?



La Placa **VORTEX SP-512** te abre un nuevo mundo de posibilidades de uso de tu ordenador.

Su instalación en el **AMSTRAD 464** se realiza en los Servicios especializados **VORTEX**, quedando bajo la garantía **PROTOMECA**.

**Ahora,  
carga a tu Amstrad  
hasta 2 Megas...**

Con los Sistemas Operativos y las Unidades de Disco **VORTEX V-DOS** estás en condiciones, no sólo de preparar y desarrollar programas más largos y complejos, sino de utilizar, de verdad, hasta 2 MEGAS de datos. El sistema operativo **VORTEX V-DOS** te lo permite. Es tan potente que, por ejemplo, puedes acceder a 16 ficheros directos, además de dos secuenciales.

Ficheros, Datos y Programas, Bases de Datos... Para dar servicio a esta capacidad de

almacenamiento y de ampliación operativa, están los Sistemas Operativos y las Unidades de Disco **VORTEX V-DOS**, listos para ser conectados a los **AMSTRAD 464, 664 y 6128**.

Acude a un Servicio **VORTEX** y descubre la nueva frontera de tu **AMSTRAD**.

Infórmate en **VORTEX**. Tel. (91) 675 75 99  
Avda. de la Constitución, 260.  
Torrejón de Ardoz. MADRID.  
O en cualquiera de los Servicios especializados **VORTEX**.

## SERVICIOS VORTEX

### ALAVA

DATAVI  
Avda. Gasteiz, 29  
Vitoria

### ALICANTE

AUDIO-COLOR  
Avda. Mazonave, 17

### AVILA

COMERCIAL ROCHA, S. A.  
C/ Arévalo, 2

### BADAJOS

DONCEL  
C/ Hernán Cortes, 3

### BALEARES

DISTELEC  
C/ Angel Guimerá, 23  
Palma de Mallorca

### BARCELONA

SVI DELEGACION CATALUÑA  
Avda. Pau Claris, 165, Piso 3º

### BURGOS

E.I.S.A.  
C/ Madrid, 4

### CADIZ

MECANOGRAFIA GADITANA  
C/ Rosario, 2

### GRANADA

TECNICAS INFORMATICAS  
APLICADAS  
Plaza Santo Cristo, 3 y 5

### GUIPUZCOA

SOFT  
C/ Cuesta de Aldapela, s/n  
San Sebastián

### DONOSTI

COMPUCARD  
Avda. de la Zurriola, 22 bis  
San Sebastián

### JAEN

OFIMATICA JAEN  
Pasaje Maza, 7

### LA CORUÑA

LOGICAR  
Urbanización Galicia, 123  
Samoedo - Sada

### LOGROÑO

EGUIZABAL C. Doctores  
Castro Viejo, 34

### MADRID

MICRO WORLD  
C/ Fernández de la Hoz, 46

### MALAGA

MANIN INFORMATICA  
Paseo Marítimo "Ciudad de  
Melilla", 25  
Piso II, Departamento C

### NAVARRA

MICRO-HOBBY  
C/ Aralar, 40  
Pamplona

### PALENCIA

LA ESFERA  
C/ Mayor Principal, 87

### SALAMANCA

QUINTA AVENIDA  
C/ España, 79

### SANTA CRUZ DE TENERIFE

EQUINTESA  
C/ San Sebastián, 74

### SANTANDER

INSUMES  
Centro Comercial  
C/ Juan Ceballos, 9, bajo  
Torrelavega

### SEVILLA

MICROTIENDA I  
C/ Aceituno, 8

### VIZCAYA

CHIP & TIPS  
Alameda de Urquijo, 63  
Bilbao

### TOLEDO

HOBBYLANDIA  
C/ Fuenlabrada, s/n  
Polígono Industrial

### VALLADOLID

CHIPS & TIPS  
Plaza de Tenerías, 11

### ZAMORA

VIDEO IMAGEN 21  
C/ San Torcuato, 21

### ZARAGOZA

DAJOL  
C/ Cánovas, 31

**vortex**  
COMPUTERSYSTEME



# DIBUJANDO

Antonio L. Aretio Najarro



Con este programa podemos realizar dibujos en pantalla y conocer las coordenadas del píxel que estamos dibujando.

Para ello nos valemos de cuatro variables (x1,x2,y1,y2) que almacenan las posiciones del píxel antes y después del movimiento.

El movimiento lo realizamos con la ayuda de ocho teclas, cuatro para movimientos horizontales y verticales y cuatro para los movimientos en diagonal. La correspondencia entre tecla y movimiento se encuentra en el folio del listado del programa.

Además, si pulsamos simultáneamente una de esas teclas y la barra de espacio, conseguiremos que el píxel se desplace por la pantalla sin dibujar.

El desglose del programa por líneas es el siguiente:

**20** Selecciona el modo de pantalla y el color del borde.

**30** Selecciona las tintas que vamos a utilizar.

**40-50** Abre una ventana en la parte inferior de la pantalla y le asigna colores al papel y a la pluma.

**60-70** Asigna los valores iniciales de las variables e imprime en pantalla un píxel de color blanco.

**89-90** Imprime en la ventana las coordenadas de la posición del píxel que estamos dibujando.

**100-170** Examina el estado de las teclas, aumentando o disminuyendo una de las dos coordenadas si alguna tecla está pulsada.

**180** Imprime en la nueva posición un píxel de color blanco.

**190** Si la barra de espacio está pulsada imprime en color negro un píxel en la posición inicial, consiguiendo de esta manera movimiento sin dibujar.

**200** Iguala las coordenadas iniciales y finales.

**210** Nos manda a la línea 80 para repetir el proceso.

El listado, así como las teclas que utilizamos son:

Y U J

H J

B N M

Barra de espacio

(Mover sin dibujar)

```
10 '=====
'      DIBUJANDO  == POR A. ARE
T10 ==
=====
20 MODE 1:BORDER 0
30 INK 0,0:INK 1,26:INK 2,6:INK 3,1
3
40 WINDOW #1,1,40,25,25
50 PAPER #1,3:PEN #1,2:CLS #1
60 x1=320:y1=200:x2=x1:y2=y1
70 PLOT x1,y1,1
80 LOCATE #1,2,1:PRINT #1,"X=";XPOS
90 LOCATE #1,20,1:PRINT #1,"Y=";YPOS
S
100 IF INKEY(42)=0 THEN y2=y2+2
110 IF INKEY(46)=0 THEN y2=y2-2
120 IF INKEY(45)=0 THEN x2=x2+2
130 IF INKEY(44)=0 THEN x2=x2-2
140 IF INKEY(43)=0 THEN y2=y2+1:x2=
x2-1
150 IF INKEY(35)=0 THEN y2=y2+1:x2=
x2+1
160 IF INKEY(38)=0 THEN y2=y2-1:x2=
x2+1
170 IF INKEY(54)=0 THEN y2=y2-1:x2=
x2-1
180 PLOT x2,y2,1
190 IF INKEY(47)=0 THEN PLOT x1,y1,
0
200 x1=x2:y1=y2
210 GOTO 80
```

**GOSLINE** s.a.  
SERVICIO INFORMATICO



## PROGRAMAS EDUCATIVOS PARA: AMSTRAD, MSX, SPECTRUM

» **EDORTO**: PROGRAMA PARA EJERCICIOS DE ORTOGRAFIA "UNA CORRECTA ORTOGRAFIA ES SINONIMO DE FORMACION Y CULTURA. Y QUE LA ALCANCE EL USUARIO PRETENDE EDORTO".

» **EDACLE**: PROGRAMA PARA EJERCICIOS DE ATENCION Y COMPRENSION LECTORA "EL OBJETIVO PRIMORDIAL DE EDACLE ES EJERCITAR Y POR TANTO DESARROLLAR, LAS CAPACIDADES DE ATENCION Y COMPRENSION LECTORA, TAN FUNDAMENTALES EN LOS ESTUDIOS COMO EN TODA ACTIVIDAD DE CARACTER INTELECTUAL".

» **EJEILE**: PROGRAMA PARA EJERCITAR EL INSTRUMENTO LECTOR "LA FINALIDAD DE EJERCICIOS LUDICO-DIDACTICOS QUE PROPONE ESTE PROGRAMA ES AGUDIZAR LA DISCRIMINACION VISUAL Y, POR TANTO, PERFECCIONAR EL INSTRUMENTO LECTOR (INSTRUMENTO LECTOR = OJOS Y CAPACIDAD DE ATENCION)".

CASSETTE..... 1.800 - DISCO..... 2.750  
3 PROGRAMAS EN DISCO O CASSETTE.... 3.500

SE NECESITAN DISTRIBUIDORES PARA TODA ESPAÑA

PEDIDOS: GOSLINE, S.A. CUARTELES, 43 - 1º, (29002) MALAGA - TEL. 31 18 77

REGALO DE UNA CALCULADORA POR CADA CASSETTE O DISCO



# GRAFICOS RELATIVOS EN C/M

*Continuaremos esta semana hablando de los gráficos en alta resolución y cómo simularlos desde código máquina, estudiando las diversas rutinas del firmware que utilizaremos para este propósito.*



El programa que hemos preparado en esta ocasión incluye los comandos gráficos que no se vieron en el anterior capítulo, y con ellos se efectúa una pantalla de presentación.

Como siempre, lo primero que debemos hacer es preparar el modo de pantalla en que se desea trabajar.

Seguidamente, y dado que vamos a trabajar con gráficos en alta resolución, deberemos colocar el cursor de gráficos en la posición adecuada, esto se consigue desde Basic, con el siguiente comando:

**MOVE X,Y**

Para conseguir el mismo resultado, se dispone de la rutina del firmware que vemos a continuación.

## **Chequea un punto en una posición absoluta #BBFO**

El cursor de gráficos se coloca en la posición indicada. Se lee el pixel que hay en esa posición y la rutina nos da la tinta que contiene dicho pixel. Si la posición dada se encuentra fuera de la ventana gráfica, nos devuelve la tinta y el papel actuales.

## **Condiciones de entrada**

El registro doble DE debe contener la coordenada X del punto a chequear, y el registro HL debe contener la coordenada Y.

## **Condiciones de salida**

El acumulador contiene la tinta especificada para el pixel que se encuentra en la posición dada. Los registros BC, DE y HL y todos los flags se corrompen, los demás registros son preservados.

Realmente nosotros utilizamos esta rutina para colocar el cursor gráfico en la posición deseada, aunque como podemos comprobar su funcionamiento es bien distinto.

La explicación de su utilización, es sencilla, ya que permite colocarnos en la posición de pantalla deseada, sin que se ilumine ningún pixel.

Una vez situados en la posición adecuada, se llama por primera vez a la rutina encargada de dibujar en pantalla una serie de cuadrados sobre una línea diagonal hacia el centro de la pantalla.

Esta rutina utiliza otros dos comandos gráficos de alta resolución, que en Basic son los siguientes:

**MOVER X,Y  
DRAWR X,Y**

este último puede llevar además dos parámetros adicionales que indiquen la tinta y la forma de impresión, y que pueden ser definidos desde código máquina de la forma que indicábamos en el anterior capítulo.

Para realizar la primera función hace una llamada a la siguiente rutina:







### Chequea un punto en una posición relativa #BBF3

El cursor de gráficos se coloca en la posición indicada. Esta rutina nos devuelve en el acumulador la tinta que contiene el pixel situado en dicha posición. Si la posición se encuentra fuera de la pantalla, se devuelven la tinta y papel actuales.

#### Condiciones de entrada

El registro doble DE debe contener el desplazamiento sobre la coordenada X, y el registro HL, el desplazamiento sobre la coordenada Y.

#### Condiciones de salida

El acumulador contiene la tinta del pixel especificado. Se corrompen todos los flags y los registros BC, DE y HL. Se preservan los restantes.

Esta última, es semejante a la estudiada anteriormente, sólo que los parámetros que se dan ahora, deben ser referidos a la posición anterior del cursor de gráficos.

Para efectuar las líneas que nos proporcionarán la formación de los cuadrados, deberemos utilizar una rutina que sea capaz de dibujarlas, como la que vemos a continuación:

### Dibuja una línea relativa #BBF9

Coloca el cursor de gráficos en el punto dado, ploteando todos los pixel que se encuentra a su paso. Los puntos que se encuentran fuera de la pantalla son ignorados.

#### Condiciones de entrada

El registro doble DE contendrá el desplazamiento sobre el eje X, y el registro HL el desplazamiento sobre el Y.

#### Condiciones de salida

Se corrompen los registros AF, BC, DE y HL. Los demás registros se preservan.

Una vez realizada la impresión de la primera serie de cuadrados, colocaremos el cursor de gráficos en la esquina inferior derecha, para volver a llamar a la rutina anterior e imprimir la segunda serie.

## Código MAQUINA

Seguidamente se coloca el cursor en las esquinas superiores y, se procede de la misma forma, con lo cual obtenemos una pantalla totalmente simétrica.

Por último para enmarcar dichos dibujos, realizaremos dos líneas horizontales y otras dos verticales, que vayan de un lado a otro de la pantalla.

Para ello colocaremos el cursor en la posición adecuada, y dibujaremos la línea correspondiente.

Desde Basic, se obtienen estas líneas mediante el siguiente comando:

DRAW X,Y

ésta también puede llevar dos parámetros adicionales, como hemos mencionado anteriormente con DRAWR.

La forma de obtener dichas líneas desde código máquina, será llamando a la siguiente rutina:

### Dibuja una línea hasta una posición absoluta #BBF9

Coloca el cursor de gráficos en la última posición dada, iluminando todos los puntos existentes entre dicha posición e inicial. Los puntos que se encuentran fuera de los límites de la pantalla, son ignorados.

#### Condiciones de entrada

El registro DE debe contener la coordenada X, y HL contendrá el valor de la coordenada Y.

#### Condiciones de salida

Se corrompen los registros AF, BC, DE y HL. Los restantes se preservan.

De esta forma colocando el cursor en cada una de la posiciones deseadas y dando los valores adecuados a cada uno de los registros, obtendremos las líneas horizontales y verticales deseadas.

Con todo lo explicado en estos dos últimos capítulos del curso, estamos en condiciones de trabajar en el modo de alta resolución de pantalla desde código máquina, sin que podamos encontrarnos con ningún tipo de problemas.



Hisoft GENA3.1 Assembler. Page 1.

Pass 1 errors: 00

```

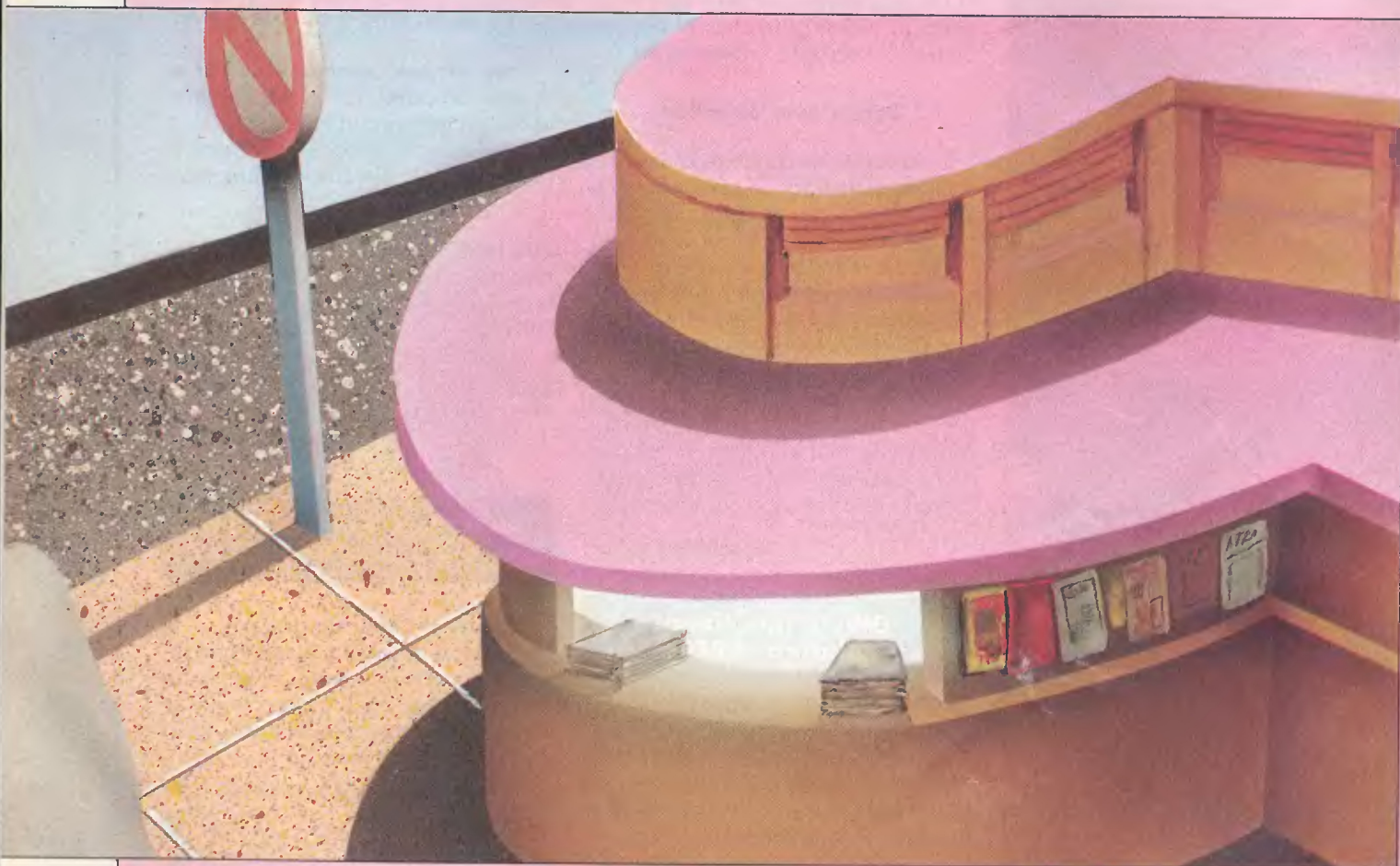
A000      10      ORG #A000
          20 ;
          30 ;MODE 1
          40 ;
A000 3E01      50      LD A,1
A002 CDOEBC    60      CALL #BCOE
          70 ;
          80 ;PLOT 180,150
          90 ;
A005 11B400   100     LD DE,180
A008 219400   110     LD HL,150

```

```

A024 CD50A0   270     CALL PLOTR
A027 CD5AA0   280     CALL PLOTS
A02A 210A11   290     LD HL,#110A
A02D 11A4A0   300     LD DE,TXT1
A030 CD97A0   310     CALL PRINT
A033 210C11   320     LD HL,#110C
A036 11ACA0   330     LD DE,TXT2
A039 CD97A0   340     CALL PRINT
A03C C9       350     RET
A03D C9       360     RET
          370 ;
          380 ;TINTA 1 PARA GRAFICOS
          390 ;
A03E 3E01     400     LD A,1
A040 CDDEBB   410     CALL #BBDE

```



```

A00B CDEABB   120     CALL #BBEA
          130 ;
          140 ;MODD DE ESCRITURA NORMAL
          150 ;
A00E AF       160     XOR A
A00F CD59BC   170     CALL #BC59
          180 ;
          190 ;BUCLE PRINCIPAL
          200 ;
A012 CD3EA0   210     CALL INK1
A015 CD5AA0   220     CALL PLOTS
A018 CD44A0   230     CALL INK2
A01B CD50A0   240     CALL PLOTR
A01E CD5AA0   250     CALL PLOTS
A021 CD4AA0   260     CALL INK3

```

```

A043 C9       420     RET
          430 ;
          440 ;
          450 ;TINTA 2 PARA GRAFICOS
          460 ;
A044 3E02     470     LD A,2
A046 CDDEBB   480     CALL #BBDE
A049 C9       490     RET
          500 ;
          510 ;TINTA 3 PARA GRAFICOS
          520 ;
A04A 3E03     530     LD A,3
A04C CDDEBB   540     CALL #BBDE
A04F C9       550     RET
          560 ;

```



```

570 ;PLOTTR 2,2
580 ;
A050 110200 590 PLOTTR: LD DE,2
A053 210200 600 LD HL,2
A056 CDEDBB 610 CALL #BBED
A059 C9 620 RET
630 ;
640 ;PRIMER BUCLE DE PLOTS
650 ;
A05A 0696 660 PLOTS: LD B,150
A05C C5 670 BUC1: PUSH BC
A05D 110000 680 LD DE,0
A060 210100 690 LD HL,1
A063 CDEDBB 700 CALL #BBED
A066 C1 710 POP BC
A067 10F3 720 DJNZ BUC1
A069 06FA 730 LD B,250
A06B C5 740 BUC2: PUSH BC
A06C 110100 750 LD DE,1
A06F 210000 760 LD HL,0
A072 CDEDBB 770 CALL #BBED
A075 C1 780 POP BC
A076 10F3 790 DJNZ BUC2
A078 0696 800 LD B,150
A07A C5 810 BUC3: PUSH BC
A07B 110000 820 LD DE,0
A07E 21FFFF 830 LD HL,-1
A081 CDEDBB 840 CALL #BBED
A084 C1 850 POP BC
A085 10F3 860 DJNZ BUC3
A087 06FA 870 LD B,250
A089 C5 880 BUC4: PUSH BC
A08A 11FFFF 890 LD DE,-1
A08D 210000 900 LD HL,0
A090 CDEDBB 910 CALL #BBED
A093 C1 920 POP BC
A094 10F3 930 DJNZ BUC4
A096 C9 940 RET
950 ;
960 ;BUCLE DE IMPRESION
970 ;
A097 CD75BB 980 PRINT: CALL #BB75
A09A 1A 990 BUC5: LD A,(DE)
A09B FEFF 1000 CP 255
A09D CB 1010 RET Z
A09E CD5ABB 1020 CALL #BB5A
A0A1 13 1030 INC DE
A0A2 1BF6 1040 JR BUC5
1050 ;
1060 ;TEXTOS
1070 ;
A0A4 414D5354 1080 TXT1: DEFM "AMSTRAD"
A0AB FF 1090 DEFB 255
A0AC 53454D41 1100 TXT2: DEFM "SEMANAL"
A0B3 FF 1110 DEFB 255

```

Pass 2 errors: 00

## TABLA DE ETIQUETAS

BUC1	A05C	BUC2	A06B	BUC3	A07A
BUC4	A089	BUC5	A09A	INK1	A03E
INK2	A044	INK3	A04A	PLOTTR	A050
PLOTS	A05A	PRINT	A097	TXT1	A0A4
TXT2	A0AC				

# Código MAQUINA

```

10 MODE 1
20 PLOT 180,150
21 X=1
30 GOSUB 100
31 X=2
40 PLOTTR 2,2
50 GOSUB 100
51 X=3
60 PLOTTR 2,2
70 GOSUB 100
81 LOCATE 17,10:PRINT "AMSTRAD"
82 LOCATE 17,12:PRINT "SEMANAL"
90 END
100 FOR N=1 TO 150
110 PLOTTR 0,1,X,0
120 NEXT
130 FOR N=1 TO 250
140 PLOTTR 1,0,X,0
150 NEXT
160 FOR N=1 TO 150
170 PLOTTR 0,-1,X,0
180 NEXT
190 FOR N=1 TO 250
200 PLOTTR -1,0,X,0
210 NEXT
220 RETURN

```

```

10 REM **PROGRAMA CARGADOR**
20 FOR N=&A000 TO &A0B4
30 READ A:SUMA=SUMA+A
40 POKE N,A
50 NEXT
60 IF SUMA<&5767 THEN PRINT "ERROR
EN DATAS"
70 DATA 62,1,205,14,188,17,180
80 DATA 0,33,150,0,205,234,187
90 DATA 175,205,89,188,205,62,160
100 DATA 205,90,160,205,68,160,205
110 DATA 80,160,205,90,160,205,74
120 DATA 160,205,80,160,205,90,160
130 DATA 33,10,17,17,164,160,205
140 DATA 151,160,33,12,17,17,172
150 DATA 160,205,151,160,201,201,62
160 DATA 1,205,222,187,201,62,2
170 DATA 205,222,187,201,62,3,205
180 DATA 222,187,201,17,2,0,33
190 DATA 2,0,205,237,187,201,6
200 DATA 150,197,17,0,0,33,1
210 DATA 0,205,237,187,193,16,243
220 DATA 6,250,197,17,1,0,33
230 DATA 0,0,205,237,187,193,16
240 DATA 243,6,150,197,17,0,0
250 DATA 33,255,255,205,237,187,193
260 DATA 16,243,6,250,197,17,255
270 DATA 255,33,0,0,205,237,187
280 DATA 193,16,243,201,205,117,187
290 DATA 26,254,255,200,205,90,187
300 DATA 19,24,246,65,77,83,84
310 DATA 82,65,68,255,83,69,77
320 DATA 65,78,65,76,255,0,0

```



**P** ara que tus dedos  
 no realicen el trabajo duro, M.H. AMS  
 TRAD lo hace por ti. Todos los listados que incluyen  
 este logotipo se encuentran a tu disposición en un cas-  
 sette mensual, solicítalo.



# Ofites Informática

**Presenta:**

**el lápiz al que gusta decir SI mientras nuestros competidores dicen no**

**UNICO PARA AMSTRAD, CON PRECISION PIXEL**

FUNCIONES	ESP	dk'tronics	OTROS
UNICO MENU DE PANTALLA	SI	NO	
ARRASTRE OBJETOS PANTALLA	SI	NO	
TRASLADO OBJETOS PANTALLA	SI	NO	
TRASLADO DE CURSOR	SI	NO	
CAJAS ELASTICAS	SI	SI	
LINEA ELASTICA	SI	SI	
TRIANGULO ELASTICO	SI	NO	
ELIPSE ELASTICO	SI	NO	
DIAMANTE ELASTICO	SI	NO	
POLIGONO ELASTICO	SI	NO	
HEXAGONO ELASTICO	SI	NO	
OCTOGONO ELASTICO	SI	NO	
CUBO ELASTICO	SI	NO	
PIRAMIDE ELASTICA	SI	NO	
CIRCUNFERENCIAS	SI	SI	
CIRCULOS RELLENOS	SI	NO	
CAJAS RELLENAS	SI	NO	
ELIPSES RELLENAS	SI	NO	
CUNAS	SI	NO	
SIMULADOR DE CORTES	SI	NO	
DISEÑO DE ZOOM	SI	SI	
IMAGEN ESPEJO E INVERTIDA	SI	NO	
FONDO DE REFERENCIA	SI	NO	
REJILLA DE FONDO	SI	NO	
OPCION DISPLAY X, Y	SI	NO	
RELLENADO CON COLOR	SI	SI	
LAVADO DE COLOR	SI	NO	
VOLCADO PANTALLA RESIDENTE	SI	NO	
DIBUJO DE BORDES EN 3 D	SI	NO	
TEXTO	SI	SI	
9 TAMAÑOS DE BROCHA	SI	NO	
18 TOBERAS MOSTRADORAS	SI	NO	
4 MEZCLAS BASICAS	SI	NO	
VARIADOR DE MEZCLAS	SI	NO	
SOMBREADO DE MEZCLAS XOR	SI	NO	
FICHERO ICONOS RESIDENTES	SI	NO	
FICHERO RELLENOS RESIDENTES	SI	NO	
26 COLORES DE PAPEL	SI	NO	
PALETA DE 15 TONOS DE COLOR	SI	NO	
POSICIONAMIENTO DE PUNTO	SI	SI	
RAYOS DESDE UN PUNTO FIJO	SI	NO	
DIBUJO REFLEJADO (ESPEJO)	SI	NO	
FUNCION HOME	SI	NO	
CONTROL DESDE TECLADO	SI	SI	
CONTROL CON JOYSTICK	SI	NO	
DISPONIBLES MODOS 1 Y 2	SI	?	

DEBIDO A LA FALTA DE ESPACIO NO PODEMOS LISTAR LAS OTRAS 40 FUNCIONES MAS QUE NUESTRO LAPIZ ES CAPAZ DE HACER.

**DISPONIBLE PARA:**

CPC 464 CASSETTE ..... 4.900 Ptas.

CPC 464-664 DISCO ..... 6.900 Ptas.

CPC 6128 DISCO ..... 6.900 Ptas.

(IVA no incluido)

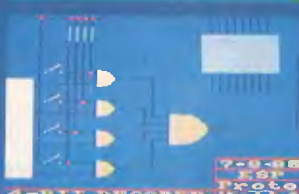
**CONDICIONES ESPECIALES PARA DISTRIBUIDORES**

**COMPARE**

THE ELECTRIC STUDIO

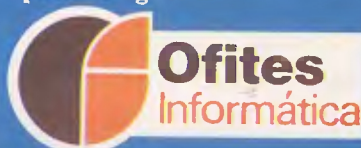
**TRADUCIDO  
AL ESPAÑOL**

**ESTOS SON  
ALGUNOS EJEMPLOS  
DE LOS GRAFICOS QUE VD.  
PODRA REALIZAR CON NUESTRO  
LAPIZ OPTICO**



**DE VENTA EN LOS MEJORES COMERCIOS  
DE INFORMÁTICA**

Si Vd. tiene alguna dificultad para obtener el lápiz óptico, puede dirigirse a:



Avda. Isabel II, 16 -8º  
Tels. 455544 - 455533  
Télex 36698  
20011 SAN SEBASTIAN



# ORDEMANIA SOFT

DE VENTAS

ESTADÍSTICA

**por fin..**

CONTABILIDAD

No. ASIENTO: 1

FECHA: 09/06/86

MANUAL

## GESPACK

Paquete integrado de gestión que le permite a Vd., de forma fácil y optimizando el tiempo, la gestión global de su empresa.

Este paquete incorpora los programas de CONTABILIDAD, FACTURACION y CONTROL DE STOCKS además de un programa de CONTROL DE PEDIDOS y todo de forma interactiva. Controle su empresa por sólo

**29.900 pts.**

### Contabilidad

Contabilidad de fácil manejo y de gran potencia que permite trabajar con cuentas de hasta cuatro niveles, con capacidad según diskette de 500/1.000 cuentas y de 2.000/10.000 asientos.

Permite modificar o dar de baja apuntes ya integrados en el Mayor, programaciones de cierres, ficheros de Contabilidad y Cuenta de Explotación, ejecución de balances comparativos, reenlazándolos por meses, clave acceso restringido, etc...

**19.900 ptas.**

### Facturación

Programa de gran agilidad y rapidez que incorpora el Control de Clientes, con gran capacidad de datos, Artículos, Albaranes, Facturas y recibos.

Generación automática y manual de documentos, valoración, a voluntad de los albaranes, todo tipo de listado, incluido el del IVA de las facturas emitidas para la declaración de Hacienda, etc...

**15.500 ptas.**

### Control de stocks

Gran capacidad de datos, le permitirán a Vd. llevar con claridad y sencillez el control de su stock.

El programa le permite llevar un libro de entradas/salidas, reorganizarlo, hacer listados de stocks..., le avisará de los límites de stocks, mínimo y máximo por artículo, etc...

Todo para la llevanza de su almacén.

**14.900 ptas.**

disponibles para: PCW 8256  
PCW 8512

Asimismo, Contabilidad disponible para CPC 664/6128 **9.900 ptas.**

**OFITES INFORMÁTICA**  
**DISTRIBUIDOR EXCLUSIVO DE ORDEMANIA SOFT**

IVA NO INCLUIDO

Condiciones especiales para distribuidores

Si tiene alguna dificultad en obtenerlos dirijase a

**Ofites**  
Informática

Avda. Isabel III, 16 - 8º  
Tels. 455544 - 455533  
Téllex 36698  
20011 SAN SEBASTIAN



# PINBALL

*Las simulaciones por ordenador cada vez abarcan un espectro más amplio en las cosas de la vida cotidiana. No sólo en los campos de gestión y técnicos, sino también en los lúdicos. En efecto, hay juegos para los que se utiliza la mecánica y la electrónica más que la informática, pero que se han hecho tan famosos y populares, tan legendarios, que tarde o temprano alguien sucumbe a la tentación de ponerlos en el escenario de un computador. Este es el caso de «Pinball», las casi tópicas «máquinas de las bolas» que tan buenos ratos han hecho pasar a casi todos los que lean este artículo, independientemente de su edad.*

Antoni Freixanet Alaña



Las teclas son redefinibles, el programa acapara 7K.

Debemos mantener la pelota el mayor tiempo posible en movimiento, para ello nos valdremos de los dos mandos para disparar la pelota, cada vez que la pelota tome contacto con una pared del circuito el marcador de puntos sumará a los ya obtenidos, 100 puntos más, si llenamos un agujero vacío sumará, 1.000 puntos y si obtenemos 20.000 puntos tendremos bola extra, si por el contrario llenamos los 15 agujeros del circuito imprimiendo en la parte inferior de la pantalla el anagrama de MICROHOBBY conseguiremos 3 bolas y 10.000 puntos.



El objetivo del juego es hacer la mayor cantidad de puntos posibles y mantener la pelota sin que se cuele.

## ESTRUCTURA DEL PROGRAMA

10	Rem del programa
50	Atributos del juego
60	Gráficos definidos
100	Variables
140	Presentación
320	Selección de controles
470	Dibuja circuito del juego
610	Disparar la pelota
770	Movimiento de mandos
840	Movimiento de la pelota
1000	Imprime puntos y mira si se gana pelota
1050	Mira si el agujero por el que entra está lleno o vacío, si está vacío lo llena, da puntos y mira si está completo
1090	Resta una pelota y mira si se han terminado
1150	Fin del juego, mira si la puntuación obtenida es mayor que el récord
1250	Datos para la música de presentación
1270	Dibuja mandos
1290	Vacía los agujeros cuando los ha llenado todos

## VARIABLES

X,Y	Coordenadas de la pelota
L	Pelotas que nos quedan
U	Marcador de puntos
T	Récord máxima puntuación
W\$	Desplaza la pelota a la derecha si es 1 y a la izquierda si es 0
E\$	Es la tecla elegida para disparar la pelota
X\$	Es la tecla elegida para mover mando izquierdo
DIM Z(15)	Es la tecla elegida para mover mando derecho
PLE	Agujeros para llenar
UP	Cantidad de agujeros llenos
I	Si es mayor que 0 la pelota sube
	Puntuación que hay que obtener para ganar una pelota



```

10 REM *****
**
20 REM ***      @ 1986 PING-BALL      *
**
30 REM *** ANTONI FREIXANET ALAIA *
**
40 REM *****
**
50 INK 0,0:INK 1,1:PEN 1:PAPER 0:PE
N 1:CLS
60 SYMBOL AFTER 253
70 SYMBOL 253,1,1,1,1,1,1,1,1
80 SYMBOL 254,0,0,24,60,60,24,0,0
90 SYMBOL 255,126,255,255,255,255,2
55,255,126
100 DIM Z(15):RECORD=20000
110 PLE=0:X=37:Y=2:S=0:T=0:L=2:U=0:
I=20000
120 FOR A=1 TO 15:Z(A)=0:NEXT
130 N$=" MICRO HOBBY "
140 BORDER 0:RESTORE:MODE 1
150 PEN 3:LOCATE 1,5:PRINT "      ###
### # # ###      ## ### # #
# # # ### #      # # # # # #
      ### # # # # # # # # ### #
# # # # # # # # # # # # #
# # # # # # # # # # # # #
### # # ### ###"
160 PAPER 2:PEN 3:FOR A=17 TO 22:LO
CATE 6,A:PRINT CHR$(143);"
      ";CHR$(143):NEXT
A:PAPER 0
161 FOR A=3 TO 35 STEP 5:LOCATE A,1
7:PRINT " ":LOCATE A,18:PRINT " ":L
OCATE A,19:PRINT " ":NEXT A
170 FOR A=6 TO 35:LOCATE A,16:PRINT
CHR$(143):NEXT A:PEN 2
180 FOR A=95 TO 554 STEP 16:PLOT A,
48:DRAWR 0,96:NEXT A
190 LOCATE 11,13:PRINT "PRESS RETUR
N TO START"
200 FOR B=1 TO 40:PEN (RND*2)+1:LOC
ATE B,25:PRINT CHR$(237);:LOCATE B,
1:PRINT CHR$(237):NEXT B
210 FOR B=2 TO 24:PEN (RND*2)+1:LOC
ATE 1,B:PRINT CHR$(237):LOCATE 40,B
:PRINT CHR$(237):NEXT B
220 PEN 2:LOCATE 2,24:PRINT CHR$(16
4);" 1986 ANTONI FREIXANET ALAIA-SA
BADELL"
230 CLEAR INPUT
240 GOTO 260
250 LOCATE NOTA1,21:PAPER 2:PRINT C
HR$(253):PAPER 0
260 READ NOTA:IF NOTA=0 THEN RESTOR
E:GOTO 260
270 NOTA1=(RND*27)+7:LOCATE NOTA1,2
1:PEN 1:PRINT CHR$(143)
280 SOUND 1,NOTA,15:SOUND 2,NOTA*2,
15:SOUND 3,NOTA*4,15
290 IF INKEY$(>CHR$(13)) THEN 250
300 LOCATE 1,25:FOR A=1 TO 26:PRINT

```

*Serie*  
**ORO**

```

:NEXT
310 A=TIME:WHILE TIME <A+50*3:WEND
320 PEN 1:LOCATE 9,8:PRINT "SELECT
CONTROLS:"
330 LOCATE 7,10:PRINT "KEY FOR SHOO
T THE BALL....";
340 A$=INKEY$:IF A$="" THEN 340
350 PAPER 2:PRINT A$:PAPER 0:W$=A$
360 LOCATE 7,12:PRINT "STICK LEFT..
.....";
370 A$=INKEY$:IF A$="" THEN 370
380 IF A$=W$ THEN 370
390 PAPER 2:PRINT A$:PAPER 0:E$=A$
400 LOCATE 7,14:PRINT "STICK RIGHT.
.....";
410 A$=INKEY$:IF A$="" THEN 410
420 IF A$=W$ OR A$=E$ THEN 410
430 PAPER 2:PRINT A$:PAPER 0:X$=A$
440 LOCATE 7,20:PRINT "PRESS 'N' FO
R REDEFINE KEYS":LOCATE 10,22:PRINT
"ANY OTHER KEY TO START"
450 A$=INKEY$:IF A$="" THEN 450
460 IF A$="N" OR A$="n" THEN CLS:GO
TO 320
470 CLS
480 FOR A=2 TO 24:LOCATE 38,A:PRINT
CHR$(143);:LOCATE 1,A:PRINT CHR$(1
43);:LOCATE 40,A:PRINT CHR$(143);:N
EXT A
490 FOR A=1 TO 40:LOCATE A,1:PRINT
CHR$(143);:LOCATE A,25:PRINT CHR$(1
43);:NEXT A
500 PEN 3:FOR B=5 TO 35 STEP 2:LOCA
TE B,4:PRINT CHR$(255);:NEXT B:PEN
1
510 FOR B=15 TO 24:LOCATE B,21:PRIN
T CHR$(143):LOCATE B,22:PRINT CHR$(
143):LOCATE B,23:PRINT CHR$(143):LO
CATE B,9:PRINT CHR$(140);:LOCATE B,
7:PRINT CHR$(131);:NEXT B
520 LOCATE 39,25:PRINT " ":PEN 2:LO
CATE 5,8:PRINT CHR$(238);"      ";CHR
$(238):LOCATE 5,12:PRINT CHR$(238);
"      ";CHR$(238):LOCATE 30,8:PRINT
CHR$(238);"      ";CHR$(238):LOCATE 3
0,12:PRINT CHR$(238);"      ";CHR$(23
8):PEN 1
530 LOCATE 25,7:PRINT CHR$(139):LOC
ATE 38,2:PRINT CHR$(138):LOCATE 18,
22:PRINT " ";CHR$(254);"2 ":LOCATE
18,16:PRINT CHR$(215);"      ";CHR$(21
4):LOCATE 15,8:PRINT CHR$(133):LOCA
TE 15,9:PRINT CHR$(141):LOCATE 15,7
:PRINT CHR$(135)
540 LOCATE 25,9:PRINT CHR$(142):LOC

```



```

ATE 25,8:PRINT CHR$(138):LOCATE 37,
20:PRINT CHR$(214):LOCATE 2,20:PRIN
T CHR$(215):LOCATE 36,22:PRINT CHR$(
214):LOCATE 36,23:PRINT CHR$(143):
LOCATE 3,22:PRINT CHR$(215):LOCATE
3,23:PRINT CHR$(143)
550 PEN 2:LOCATE 35,20:PRINT CHR$(2
31):LOCATE 4,20:PRINT CHR$(231):GOS
UB 1270:GOSUB 1280
560 LOCATE 24,21:PRINT " ":LOCATE 1
5,21:PRINT " ":LOCATE 23,21:PRINT C
HR$(215):LOCATE 24,22:PRINT CHR$(21
5):LOCATE 16,21:PRINT CHR$(214):LOC
ATE 15,22:PRINT CHR$(214)
570 PEN 3:LOCATE 18,14:PRINT CHR$(2
30);" ";CHR$(230);" ";CHR$(230):PEN
1:LOCATE 18,17:PRINT CHR$(131);CHR
$(131);" ";CHR$(131);CHR$(131)
580 LOCATE 15,24:PRINT CHR$(143):LO
CATE 16,24:PRINT USING "#####";R
ECORD;:PRINT CHR$(143)
590 LOCATE 3,15:PRINT CHR$(222);CHR
$(223):LOCATE 3,16:PRINT CHR$(221);
CHR$(220)
600 LOCATE 35,15:PRINT CHR$(222);CH
R$(223):LOCATE 35,16:PRINT CHR$(221
);CHR$(220)
610 Y=24:X=37:CLEAR INPUT
620 IF INKEY$="" THEN 620
630 IF INKEY$<>W$ THEN 630
640 LOCATE 39,24-Q:PRINT " "
650 Q=Q+1:IF Q=10 THEN Q=0
660 LOCATE 39,24-Q:PRINT CHR$(254);
670 Q1=100+Q*5
680 SOUND 1,Q1,3
690 FOR N=1 TO 9:A$=INKEY$:IF A$=W$
THEN 640
700 NEXT N
710 Y=24-Q
720 LOCATE 39,Y:PRINT " ":Y=Y-1:LOC
ATE 39,Y:PRINT CHR$(254);
730 FOR N=1 TO 30:NEXT N:IF Y>2 THE
N 720
740 Q=Q*3:LOCATE 39,2:PRINT " "
750 LOCATE X,2:PRINT " ":X=X-1:LOCA
TE X,2:PRINT CHR$(254);
760 FOR N=1 TO 40:NEXT N:Q=Q-1:IF Q
>0 THEN 750
770 IF Y=24 THEN 1090
780 IF P1=1 THEN GOSUB 1280:LOCATE
5,22:PRINT " ":LOCATE 6,21:PRINT "
":LOCATE 26,22:PRINT " ":LOCATE 27,
21:PRINT " ":P1=0:IF (Y=23 OR Y=22
OR Y=21) AND ((X>3 AND X<8) OR (X>2
4 AND X<29)) THEN UP=30:SOUND 1,200
,1
790 IF P2=1 THEN GOSUB 1270:LOCATE
13,22:PRINT " ":LOCATE 12,21:PRINT
" ":LOCATE 34,22:PRINT " ":LOCATE 3
3,21:PRINT " ":P2=0:IF (Y=23 OR Y=2
2 OR Y=21) AND ((X>10 AND X<15) OR

```

```

(X>31 AND X<36)) THEN UP=30:SOUND 1
,200,1
800 A$=INKEY$:PEN 2
810 IF A$=E$ THEN CLEAR INPUT:LOCAT
E 5,23:PRINT " ":LOCATE 26,23:PRI
NT " ":LOCATE 5,22:PRINT "0":LOCA
TE 6,21:PRINT "0":LOCATE 26,22:PRIN
T "0":LOCATE 27,21:PRINT "0":P1=1:P
P1=2
820 IF A$=X$ THEN CLEAR INPUT:LOCAT
E 11,23:PRINT " ":LOCATE 32,23:PR
INT " ":LOCATE 13,22:PRINT "0":LO
CATE 12,21:PRINT "0":LOCATE 34,22:P
RINT "0":LOCATE 33,21:PRINT "0":P2=
1
830 PEN 1
840 LOCATE X-1,Y:A$=COPYCHR$(#0):LO
CATE X+1,Y:B$=COPYCHR$(#0)
850 IF Y=4 AND X>4 AND X<35 THEN PA
PER 2:LOCATE X,Y:PRINT " ":PAPER 0:
GOSUB 1050:GOTO 880
860 LOCATE X,Y:PRINT " "
870 RN=INT (RND*10):IF RN=3 THEN 93
0
880 IF T=0 AND A$="" THEN X=X-1:GO
TO 930
890 IF A$<>" " THEN GOSUB 1000
900 T=1
910 IF T=1 AND B$="" THEN X=X+1:GO
TO 930
920 T=0:GOSUB 1000
930 LOCATE X,Y-1:C$=COPYCHR$(#0):LO
CATE X,Y+1:D$=COPYCHR$(#0)
940 IF UP=0 AND D$="" THEN UP1=UP1
+1:Y=Y+1:GOTO 980
950 IF D$<>" " AND UP=0 THEN UP=INT
(UP1/2)
960 IF C$<>" " THEN UP=0:UP1=0:GOSU
B 1000:GOTO 980
970 IF UP>0 THEN Y=Y-1:UP=UP-1:IF U
P=0 THEN UP1=0
980 PEN 3:LOCATE X,Y:PRINT CHR$(254
);:PEN 1
990 GOTO 770
1000 U=U+100
1010 IF U>=I THEN I=I+20000:BORDER
2:SOUND 1,200,5:L=L+1:LOCATE 19,22:
PRINT L:LOCATE 19,22:PRINT CHR$(254
)
1020 LOCATE 16,8:PRINT USING "####
###";U
1030 SOUND 1,20,2:SOUND 2,200,2:SOU
ND 3,400,2
1040 BORDER 0:RETURN
1050 P=(X/2)-2:IF Z(P)=0 THEN Z(P)=
1:U=U+1000:PLE=PLE+1:GOSUB 1020
1060 PAPER 2:LOCATE X,25:PRINT MID$(
N$,P,1):PAPER 0
1070 IF PLE=15 THEN GOSUB 1290
1080 RETURN
1090 FOR B=100 TO 500 STEP 10:SOUND

```



# Serie ORO



```

1,B,3:SOUND 2,B+100,3:NEXT B
1100 L=L-1:IF L=-1 THEN 1150
1110 LOCATE X,Y:PRINT " "
1120 LOCATE 19,22:PRINT L:LOCATE 19
,22:PRINT CHR$(254)
1130 PEN 1:LOCATE 39,24:PRINT CHR$(
254)
1140 GOTO 610
1150 LOCATE 16,12:PRINT "GAME OVER"
1160 FOR B=0 TO 300
1170 SOUND 1,B,1,12: BORDER B/12
1180 NEXT B
1190 CLEAR INPUT
1200 IF INKEY$="" THEN 1200
1210 LOCATE 1,25:FOR A=1 TO 26:PRIN
T:NEXT
1220 A=TIME:WHILE TIME <A+50*5:WEND
1230 IF U>RECORD THEN RECORD=U:LOCA
TE 16,24:PRINT USING "#####";REC
ORD
1240 GOTO 110
1250 DATA 142,142,142,142,159,179,1
90,213,213,179,142,106,106,106,106,
119,134,134,142,159,159,142,134,142
,134,142,119,134,142,142,159,179,17
9,190,213,190,190,190,190,179,190
1260 DATA 0
1270 PEN 2:LOCATE 11,23:PRINT "0000
":LOCATE 32,23:PRINT "0000":PEN 1:R
ETURN
1280 PEN 2:LOCATE 4,23:PRINT "0000"
:LOCATE 25,23:PRINT "0000":PEN 1:RE
TURN
1290 FOR N=1 TO 500:INK RND*1,RND*2
6:SOUND 1,N,1,7:NEXT N:U=U+10000:GO
SUB 1020:L=L+3:LOCATE 19,22:PRINT L
:LOCATE 19,22:PRINT CHR$(254):FOR N
=6 TO 34 STEP 2:LOCATE N,4:PRINT "
":LOCATE N,25:PRINT CHR$(143):NEXT
N:FOR N=1 TO 15:Z(N)=0:NEXT N
1300 Y=Y+1:PLE=0:INK 0,0:INK 1,1:RE
TURN

```



**P** ara que tus dedos  
no realicen el trabajo duro, M.H. AMS  
TRAD lo hace por ti. Todos los teclados que incluyen  
este logotipo se encuentran a tu disposición en un cas-  
sette mensual, solicitándolo.



# Sin duda alguna

A través de esta sección se pretende resolver, en la medida de lo posible, todas las posibles dudas que «**atormenten**» a todas las personas interesadas en el mundo del AMSTRAD, sean o no poseedores de uno y, si lo son, se encuentren en cualquier nivel de destreza en su manejo.

Semanalmente, aparecen en estas páginas las consultas de la mayor cantidad de usuarios posible; ello redundará en un mejor servicio y en un contacto más estrecho entre todos nosotros a través de la revista.

**SIN DUDA ALGUNA** está abierta a todos.

## RECUPERACION DE FICHEROS BORRADOS

Quisiera saber si por error borro un programa como era, «**PROGRAMA BAS**» si existe la posibilidad de recuperarlo. He leído que al borrar algo del disco, éste sólo lo hace en el directorio, y que el programa sólo se elimina cuando se graba algo encima. Si esto es así, ¿me podríais indicar cómo poderlo recuperar?

También me gustaría saber si **Indescomp** tiene la intención de traducir al castellano los programas de utilidades (que no sería mala idea) y que se regalan con el **CPC 128**, ya que en el manual no existe mucha información al respecto.

**Rafael**

1) En principio, no existe tal posibilidad. Si lo borras, lo borras. No obstante, como muy bien comentas, el ordenador lo que hace es borrar el nombre y los datos del fichero en cuestión del directorio solamente, es

decir, el fichero sigue existiendo físicamente en el disco. El problema es que, para el **Amstrad**, el espacio ocupado por el fichero difunto es utilizable, y puede que escriba encima sin la menor consideración.

Resumiendo, para recuperar un fichero del disco borrado necesitas un programa especial de utilidad que lo haga, tipo «**ODDJOB**» o algo así, pero si, en el interior, se ha escrito algo encima del fichero a recuperar, no hay nada que hacer.

2) No sabemos nada de las intenciones del **Indescomp**; lo mejor sería que les preguntaras a ellos, refiriéndote en concreto al programa que te interese.

## COMANDOS RSX

Les escribo esta carta dirigida a su apartado Sin duda alguna.

He leído el artículo del n.º 10 respecto al **RSX** y a pesar de que mis conocimientos de código máquina no son muy avanzados que digamos, lo he entendido bastante bien.

La duda que tengo es cómo se haría un comando en que el parámetro fuese una expresión. Por ejemplo en **Basic** tenemos **SQR** (expresión) para hallar la raíz cuadrada de un número o expresión, como **x+2** o **COS(x)** (dándole valores a x). ¿Cómo se haría este comando en el **RSX**?

**Fernando Goyeneche**

En cualquier comando **RSX**, se puede dar como valor cualquier parámetro, tal como **x**, **x+2**, etc. Podemos, por lo tanto, introducir un comando de la siguiente forma:

**IRSX,X,Y**

así pues, creemos quedará resuelta totalmente la duda.

## LOCOMOTIVE BASIC Y VARIABLES

Aprovecho esta carta para felicitaros por la excelente revista que estáis realizando y para ver si me podéis solucionar unos pequeños problemas:

¿Cómo almacena el **Basic Locomotive** las variables? ¿Con qué criterio lo hace (caracteres, coma flotante, BCD)? ¿Realiza las operaciones aritméticas mediante logaritmos? ¿Dónde están las llamadas «**variables del**

**sistema**» y dónde se puede conseguir un listado de ellas con sus funciones? ¿Se puede, por software, conmutar los bloques de 16K? ¿Cómo?

¿Cómo se puede listar la ROM del controlador del/de los disco/s?

**D. S. Barrientos**

Intentaré responder a tus preguntas ordenadamente.

1.º El **Locomotive Basic**, almacena las variables alfanuméricas como cadenas de caracteres, y las numéricas, en coma flotante.

2.º No realiza las operaciones aritméticas mediante logaritmos.

3.º Las Variables del sistema se encuentran a partir de la dirección hexadecimal #A900, aunque no existe ninguna información sobre ellas.

4.º Se pueden conmutar los bloques de 16 K, y como habrás podido observar ya se ha publicado un programa que lo hace en los últimos números de nuestra revista.

5.º Dicha ROM se puede listar con la ayuda de un monitor-desensamblador.

## MEMORY FULL

A continuación les paso una consulta. Tengo un **CPC 6128**, pero me encuentro que la mayoría de programas en cinta dan error de **MEMORY FULL**. ¿No podrían sacar y publicar un programa que anularse el disco, para poder introducir estos juegos?

He introducido el programa **LOGO** y me sale en pantalla «disco protegido». He hablado con otros dos poseedores de 6128 y les ocurre lo mismo, he recurrido al programa **HELP**, pero éste está escrito en inglés, por lo que sigo como estaba, ¿podríais orientarme?

**José Manuel Conde**

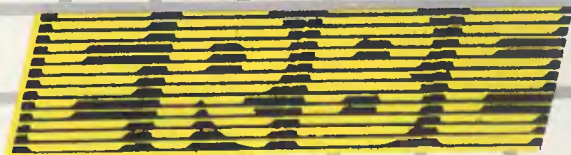
Tu sugerencia de publicar un programa que anule el disco para permitir cargar en el 6128 los juegos del 464 está en proyecto, y muy pronto podremos publicar un completo programa al respecto, a pesar de las dificultades que entraña.

De momento, existe un programa publicado en el n.º 1 de la revista «**AMSTRAD USER**» que te servirá de gran ayuda.

En cuanto a tus problemas con el **LOGO**, debes realizar una copia de seguridad del disco original antes de ejecutar el programa; de lo contrario no te funcionará.



# TU PUEDES FORMAR PARTE DE



¡Como lo oyes! Tú puedes estar informado antes que nadie de todas las novedades que aparecen en el mercado, tener acceso a programas exclusivos y a precios especiales, poder conseguir los trucos que te ayuden a salir de esa pantalla que se te resiste, regalos, pósters, sorteos, pegatinas... y un montón más de ventajas.

## APUNTATE YA AL CLUB

No lo pienses más. Si te apuntas ahora al Club ERBE vas a recibir, además, como regalo de inscripción:

► El juego que más te guste de nuestro catálogo\*



► Una fabulosa camiseta con el anagrama del Club.



► Tu carnet personal de socio, que te dará derecho al 10% de descuento en todos los juegos de ERBE que compres, durante un año, en cualquiera de los establecimientos que aquí al lado se relacionan (o a través del Club si en tu zona no existiera ninguno).



CARNET VACILON  
PARA COMPRAR UN 10%  
MAS BARATO



JUEGO FLIPANTE

CAMISETA MOLONA

**TODO POR SOLO 3.000 PTS.... UN AUTENTICO CHOLLO**

Elige ya tu juego, corre al teléfono y llámanos al (91) 447 34 10 o escríbenos a ERBE SOFTWARE C/. Santa Engracia, 17 - 28010 Madrid para hacerte socio o pedir más información.

\* QUEDAN EXCLUIDOS LOS PROGRAMAS EN DISQUETE



## DISTRIBUIDORES AFILIADOS

**ALAVA**  
DATAVI. Avda. Gasteiz, 29. VITORIA

**ALMERIA**  
INFORMATICA-ELECTRONICA. Arapiles, 22.

**ALICANTE**  
INFOTRONICA. Doctor Jiménez Díaz, 2. ELCHE.  
MICRO CENTRO. César Ezquezábal, 45. ALICANTE.  
MULTISYSTEM. San Vicente, 53. ALICANTE.  
CODE-2000. Ramón y Cajal, 3. DENIA.  
SILICON VALLEY. Girona, 4, semicéntrico. ELCHE (Alicante).

**ASTURIAS**  
CASA-RADIO MIERES. Jerónimo Ibrán, 11. MIERES.  
SOVIELECTRONICA. Cabrales, 31. GIJÓN.  
BERNE. Menéndez Valdés, 13. GIJÓN.  
RADIO-NORTE. Uña, 20. OVIEDO.  
IMAGEN. Pablo Iglesias, 83. GIJÓN.  
COMERCIAL ARANGO. Marcos de Termello, 2. AVILES.  
CUADRADO INFORMATICA. Toreno, 5. OVIEDO.

**AVILA**  
DISCO-70. Plaza Sta. Teresa, 1.

**BADAJOS**  
SONIDO RUBIO. Avda. Fdez. Calzadilla, 10. BADAJOZ.  
RADIO GRAJERA. San José, 11. ALMENDRALEJO.

**BALEARES**  
ERGON. Falangista Laportilla, 2. P. MALLORCA.  
PROCONT. Extremadura, 31. IBIZA.  
COMPUSHOP. Vía Alemania, 11. P. MALLORCA.  
TEKNOS. Aragón, 30. P. MALLORCA.

**CADIZ**  
PARODI-DISCOS. Novena, s/n.

**CASTELLON**  
CASO, S. A. San Vicente, 6. Avda. Rey Don Jaime, 74.  
APARATOS. Mayor, 32. VINAROS.

**LA CORUÑA**  
PHOTOCOPY. Juana de Vega, 29-31.

**GERONA**  
CENTRAL FOTO. Ctra. de San Feliú, 28. PLATJA D'ARO.

**GRANADA**  
INFORMATICA-ELECTRONICA. Melchor Almagro, 8.

**GUADALAJARA**  
ABI. Padre Félix Flores, 3.

**GUIPUZCOA**  
SABA. Fuenterrabía, 14. SAN SEBASTIAN.  
AMASONIK. Paseo Colón, 80-82. IRUN.

**HUELVA**  
RADILUX. Concepción, 6.

**LEON**  
MICRO BIERZO. Carlos I, 2. PONFERRADA.

**LOGROÑO**  
COMPUTER PAPEL. Castroviejo, 19.

**LUGO**  
MED INFORMATICA. Avda. Ramón Ferrero, s/n.

**MADRID**  
HIESA INFORMATICA. Camino Vinateros, 40.  
INSTRUMENTOS MUSICALES ANGEL. Plaza España, 2 (local 9). LEGANES.  
COMPUTIQUE. Embajadores, 90.

**MALAGA**  
TODO INFORMATICA. Avda. Aurora, 14.  
INFORMATICA EUROPA. Moreno Carbonero (Edif. Carbonero). FUENGIROLA.  
ORGANIZACION EMPRESAS. Ricardo Soriano, 35. MARBELLA.  
TELEVISION PIÑAS. Dr. Eusebio Ramírez, 2. SAN PEDRO ALCANTARA.

**MURCIA**  
MEMORY SHOP. Lepanto, 1.

**NAVARRA**  
MICROORDENADORES RAMAR. Navarros Villoslada, 7.  
PAMPLONA.

**ORENSE**  
ALMACENES MENDEZ. Capitán Cortés, 17.

**PALENCIA**  
LA ESFERA. Mayor, 87.

**SANTANDER**  
RADIO MARTINEZ. Doctor Jiménez Díaz, 13.

**TOLEDO**  
CALCO. Angel de Alcázar, 56. TALAVERA DE LA REINA.

**VALENCIA**  
RADIO COLON. Colón, 7.  
ELECTRONICA MORANT. Jaime Torres, 12. GANDIA.

**VALLADOLID**  
CHIPS AND TIPS. Plaza Tenebría, 11.  
MICROLID. Gregorio Fernández, 6.

**VIZCAYA**  
REMBAT. General Concha, 12. BILBAO.  
EPROM 2. Juan XXIII, 3. SANTURCE.

**ZARAGOZA**  
ADA COMPUTER. Independencia, 24-26.

**ERBE**

## Mercado común

Con el objeto de fomentar las relaciones entre los usuarios de AMSTRAD, **MERCADO COMUN** te ofrece sus páginas para publicar los pequeños anuncios que relacionados con el ordenador y su mundo se ajusten al formato indicado a continuación.

En **MERCADO COMUN** tienen cabida, anuncios de ventas, compras, clubs de usuarios de AMSTRAD, programadores, y en general cualquier clase de anuncio que pueda servir de utilidad a nuestros lectores.

Envíanos tu anuncio mecanografiado a: **HOBBY PRESS, S.A.**  
**AMSTRAD SEMANAL.**  
Apartado de correos 54.062  
28080 MADRID  
**¡ABSTENERSE PIRATAS!**

**Club de usuarios Amstrad en Valencia.** Ya tenemos un nuevo club de amigos del **Amstrad** en Valencia. Ponte en contacto con nosotros mediante el teléfono: 375 77 02 (preguntar por Vicente) y 375 41 43 (preguntar por José Luis) o mandando una carta con tus señas y teléfono a: Vicente Bresó Flores. Julio Colomer 5-1. Alfajar (Valencia).

**Vendo cables para ampliación de monitor, alimentación y antena según medidas pedidas, para más información escribir a:** Manuel Antonio Ramos Romero. C/ Rodríguez Marín, 42. San Juan de Aznalfarache (Sevilla).

**Desearía contactar con usuarios de Amstrad 464, para cambiar algunos juegos. Llaman al tel. 72 47 21. C/ Villalar, 3, Dos Hermanas (Sevilla).**

**Microhobby Amstrad Semanal.** Comprado en febrero de 1986. Buen precio. Interesados llamar al teléfono: (93) 201 24 07. Preguntar por Javier. En horas de comida o cena.

**Vendo Amstrad 6128 con monitor en color y muchos extras.** Todo comprado en marzo. Precio a convenir. Tel. 250 18 95.

**Vendo Amstrad 664 con monitor en color.** Precio 90.000 pesetas. Preguntar por Fernando en el tel. (945) 25 52 68. C/ Pintor Pablo Uranga, 20-1.º Vitoria.

**Compro Amstrad CPC 664 ó CPC 6128 con monitor en fósforo verde, que estén en buen estado.** Podéis llamar al (93) 372 58 68 de Barcelona, preguntando por Francesc, a partir de las 21 horas.



AMSTRAD

**¡NO HAY COMPETENCIA POSIBLE!**  
en Calidad / Precio



**750**  
pts.



AMSTRAD  
COMMODORE



MSX  
AMSTRAD  
SPECTRUM

MSX  
AMSTRAD  
SPECTRUM



KNIGHT TYME

La tercera de la serie de aventuras Magic Knight, se encontró a sí mismo transportado al siglo 25 abordo de la nave estelar PISCIS. El juego utiliza un sistema mejorado de animación, que fue utilizado por primera vez en Spellbound. ¿Será éste el fin de Magic Knight?.

SERIE M.A.D.: P.V.P. 1.100 PTAS.

AMSTRAD



RADZONE

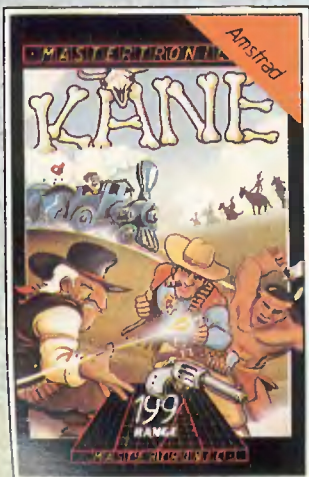
El equipo de colonización casi había terminado su tarea cuando un reactor nuclear alcanzó un estado crítico y la base tuvo que ser abandonada. Tú estas abajo todavía para volver a hacer el planetoido seguro.



THE LAST V-8

En cualquier otro coche no tendrías ninguna oportunidad; en el último V-8 la supervivencia es posible (a lo mejor ...) Sintetizador de voz, perfecto Scroll de 360°, tema musical a tres voces, gráficos futuristas excelentes.

AMSTRAD  
COMMODORE



KANE

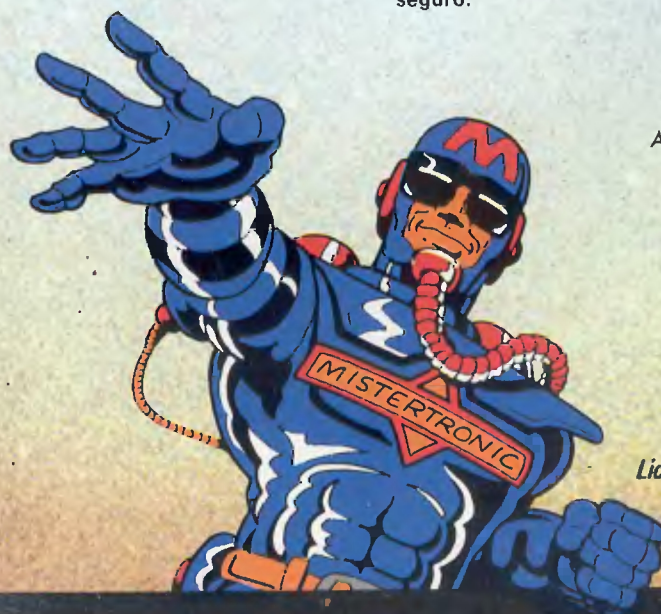
En el Oeste no caben las palabras, sino la Acción, la Rapidez y la Puntería.



INTO OBLIVION

Después de la destrucción del vil ordenador que gobernó una vez en la Galaxia, debes buscar el único planeta seguro para evitar a los sádicos andróides que han sido programados para volatizarte. Con 2.500 pantallas, tu tarea no es nada fácil.

AMSTRAD



Licencia exclusiva para ESPAÑA DRO SOFT  
Fundadores, 3 - 28028-MADRID  
Tels. 255 45 00/09



# POTENCIA PARA SU AMSTRAD



Haga su pedido por teléfono. Entregas en 24 horas



Nuestros productos se encuentran en las mejores tiendas de informática. Si tienen problemas en obtenerlos, contacten directamente



SOFTWARE y PERIFERICOS

Importador en exclusiva de los productos

**dktronics**

**COMERCIAL HERNAO, S. A.**

Serrano, 30 - 3.ª  
Teléfono (91) 435 67 64 (4 líneas)  
Telex 47340 NAO E  
28001 MADRID

## DISTRIBUIDORES:

VALENCIA (MALIOT, S. L. TEL.: (96) 367 94 61  
ZARAGOZA (BAZAR CANARIAS TEL.: (976) 23 74 90)

BARCELONA (SUMINISTROS VALL PARADIS, TEL.: (93) 381 65 70)

En los Departamentos **Online** de **GALERIAS**